VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Sociální sítě v oblasti spolupráce vývojářů
Social Network and Partnership of Developers

2011

Alisa Babskova

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

Ostrava, 5 května, 2011 _____

Abstrakt

V současné době existuje velké množství systému, které spojují různé komunity lidí a jejich aktivity. S ohledem na rozsáhlost, počet uživatelů a informací v takových systémech, je stále aktuálnější problém najít vhodné lidi pro spolupráci, nebo skupiny lidí v internetové síti. Tento problém je jedním z důvodu vzniku sociálních sítí na internetu. V této práci se budeme zabývat metodami analýzy sociálních sítí vytvořených z rozsáhlých systémů pro spolupráci vývojářů. Jako příklad byl vybrán systém www.codeplex.com. V práci si také ukážeme význam centralit grafu sítí pro analýzu komunit a týmů v sociálních sítích.

Klíčová slova: **sociální sít, analýza sociální sítě, graf, centrality**

Abstract

Currently, there are many systems that link various communities of people and their activities. Given the extensiveness, the number of users and information in such systems is still a topical problem to find suitable people or team of people across the Internet. This problem is one of the reasons of rise social networks on the Internet. In this work we will deal with methods of analysis of social networks made up of large communications systems developers. This is an example of www.codeplex.com. We'll also to demonstrate sense of centrality of graph networks for analyzing communities and teams in social networks.

Keywords: **Social Network, Social Network Analyze, Graph, Centralities**

Obsah

Seznam obrázků

Seznam tabulek

Seznam příloh

1. Úvod

V současné době existuje velké množství systému, které spojují různé komunity lidí a jejich aktivity. S ohledem na rozsáhlost, počet uživatelů a informací v síti internet, je stále aktuálnější problém nejen s vyhledáváním informací ale i vhodných lidí (uživatelů) v internetové síti. Tento problém je jedním z důvodu vzniku sociálních sítí na internetu. Sociální sítě si můžeme představit jako graf, kde uzly jsou účastníci sociální sítě a hrany jsou spojení mezi nimi. Spojením mezi uživateli, může být například sdílení společných zájmů.

Současné populární sociální sítě, se vytváří cestou registrování uživatelů a postupného přidání dalších kontaktů kolem těchto uživatelů. Při hlubším pohledu, do ostatních již existujících systémů v internetové síti jsme zjistili, že komunity lidí existují i jinde. Fóra, portály, systémy určené pro studenty, zaměstnance, nebo pro lidi s různými společenskými zájmy. Pomocí takových internetových systémů vznikají různé komunity lidí, které se taky dají pojmout jako sociální sít, jen s tím rozdílem, že na první pohled není ihned odhalitelný její graf.

Cílem této diplomové práce, byl průzkum současného stavu v problematice sociálních sítí v systémech pro sdílení informací. Dále studium metod analýzy sociálních sítí a implementace knihovny pro počítání centralit nad sociální sítí. Jedním z hlavních cílů bylo také hledání způsobů analýzy týmové spolupráce vývojářů a to na příkladů systemu pro sdílení zdrojových kódu a projektů www.codeplex.com.

1.1 Struktura práce

Práce je rozdělena, do devíti kapitol. Hned za úvodem následuje kapitola 2, která odpovídá na otázku, co je sociální síť. V kapitole 3, se dočteme o historii sociálních sítí. V kapitole 4, budou popsány existující metody analýzy sociálních sítí a význam centralit grafu sítě pro tuto analýzu. Kapitola 5, je zaměřená na vysvětlení aspektů týmové spolupráce vývojářů v rámci sociálních sítí. V kapitole 6, je popsaná příprava vstupních dat pro experimenty. V kapitole 7 si popíšeme algoritmus vytvoření sociálních sítí z dat, které jsme připravili z projektu www.codeplex.com.

V kapitole 8 ukážeme výsledky experimentů nad vytvořeným grafem sítě, popíšeme výsledný graf a jeho vlastnosti. Pomocí výpočtu centralit grafu sítě, ohodnotíme vlastnosti nalezených komunit v grafu. Dále popíšeme strukturu těchto komunit a ukážeme filtrování celého grafu sítě.

V kapitole 9 provedeme celkové shrnutí práce a vyhodnotíme provedené experimenty.

2. Pojem sociální sítě

Sociální síť - je propojená skupina lidí, kteří se navzájem ovlivňují. Sociální síť se tvoří na základě zájmů, rodinných vazeb nebo z jiných důvodů. Tento pojem se dnes také často používá ve spojení s internetem a nástupem webů, které se na vytváření sociálních sítí přímo zaměřují. Sociální sítě se můžou vytvářet také v zájmových komunitách kolem určitých webů, například na jejich fórech [19]

Z pohledu informačních systémů sociálních sítí, nejsou v podstatě ničím jiným než kombinací specializované webhostingové služby a specializovaného vyhledávače. Uživatel si vyplní svůj strukturovaný profil a hned poté může hledat a být vyhledáván. Na rozdíl od klasických vyhledávačů má uživatel sociální sítě k dispozici strukturovaná data v přesně stanoveném formátu a položky kategorizované do přesných číselníků. Například uživatel má možnost najít bývalé kolegy, kteří s ním pracovali ve stejné firmě v určité době nebo odborníka v určitém oboru, dále rodinné příslušníky nebo kamarády pobývající v zahraničí. Lze tedy vyhledávat přátele a známé, ale i nacházet nové kontakty, které by nám mohly být užitečné. Navíc se s nalezenými lidmi můžeme propojit, což znamená, že si navzájem zpřístupníme své kontakty. S tím, jak se počet našich přímých kontaktů zvyšuje, je pro nás systém stále užitečnější, neboť můžeme vidět i prohledávat stále větší síť.

Všeobecné sociální sítě mají zpravidla co nabídnout téměř jakémukoli uživateli a především umožňují registraci jakéhokoli uživatele bez rozdílu. Příkladem takovéto sociální sítě je například Facebook [18] .
Oborové sociální sítě zahrnují sítě, ve kterých se sdružují uživatelé, zabývající se stejným oborem, ať už na profesionální nebo na zájmové či studijní úrovni. Tyto sítě vznikají často okolo webových stránek s příslušnou tématikou. Základními typy sociálních sítí s ohledem na odbornost jsou [18] .

- Profesionální sociální sítě. Sítě tohoto typu sdružují profesionály daného oboru, které často nebývají anonymní nebo stoprocentně otevřené všem zájemcům. Tyto sociální sítě jsou specificky navrženy pro určité profesní či zájmové skupiny.
- „Hobby“ sociální sítě. Tato skupina zahrnuje sociální sítě, které sdružují uživatele zabývající se jistou problematikou na hobby úrovni.
- Studentské sociální sítě. Sociální sítě tohoto typu nemusí sdružovat pouze studenty, ale zaměřují se na studium (buď nějakého konkrétního oboru, případně více oborů – potom se může jednat například o studentskou komunitu okolo nějaké univerzity).
- Nezaměřené/všeobecné sociální sítě. Sítě tohoto typu nejsou zaměřeny na žádný specifický obor, případně je však v jejich rámci uživatelům umožněno sdružovat se individuálně do tematických skupin

Tyto systémy fungují dobře a jsou stále populárnější. V důsledku toho významně narůstá počet jejich uživatelů. A v důsledku toho nastává zajímavý jev: sociální sítě začínají tvořit důležitou část internetového obsahu a dá se říct, že se tak internet stává organizovanějším místem k životu.

3. Historie

Pojem „sociální síť“ byl zaveden v roce 1954 sociologem z „Manchesterovy školy“ Jamesom Barnsom, dlouho před tím, než vznikl internet a všechny současné internetové sítě [20] .

Dnes pojem sociální síť znamená určité okolí člověka. Samotný člověk je uzlem sociální sítě, kde jeho známí jsou jeho sousední uzly a vztahy mezi lidmi jsou vztahy i v sociální síti. Hlubší průzkum nad sociálními sítěmi nás navede na klasifikace vztahů mezi objekty: jednosměrné, obousměrné, sítě kolegů, spolužáků, aj.

Ve druhé půlce 20. století se sociální sítě začaly vyvíjet jako vědecká koncepce. Stalo se to, že se tento obyčejný pojem sociologů stal trendovou koncepcí, která se stala jedním ze základních pilířů koncepce WEB 2.0, o které se jako první zmínil Tim O'Reilly v září roku 2005 ve svém článku „Tim O'Reilly – What is Web 2.0“ [17] .

Z obecného hlediska sociální sítě, byly nedělitelnou současí života lidí již mnohem delší dobu a to pod pojmem sociálních skupin společnosti v které žijeme.

Sociální skupina je v nejširším smyslu jakékoliv lidské seskupení jedinců zpravidla od počtu tří (někdy i dvou), až k velkým celkům. Je to sociální formace, která se skládá z určitého množství osob, které jsou k sobě ve vzájemných vztazích – na jejichž základě na sebe působí a ovlivňují své pozice (hierarchie) a role [17] .

Z hlediska informačních technologii první sociální sítě tvořily skupiny lidí, které používaly klasické maily pro podporu svých sociálních vztahů. Stalo se tak 02.10.1971 v den, kdy byl odeslán první vzkaz na vzdálený počítač. Prvními uživateli sociální sítě se stali „vojáci“ v síti ARPA NET.

Dalším krokem bylo objevení IRC (Internet Relay Chat – chat přes internet) - systému pro komunikaci v reálném čase. IRC je vynález finského studenta Jarko Ojkarinnen . Již v roce 1988 vznikaly velké sociální sítě, které se nemohou s dnešními srovnávat.

První počítače, elektronická pošta, IRC a mnoho dalších technologií se fakticky proměnili v to, čemu dnes říkáme internet. 7. srpna 1991 britský vědec Tim Berns-Lee jako první publikoval internetové stránky a udělal tím další krok k vzniku sociálních sítí tak, jak je známe dnes.

Pokud dnes mluvíme o sociální síti, tak se většinou jedná o sociální internetovou síť, sociální web, což v podstatě neznamená nic jiného, než webové stránky založené a určené hlavně pro komunikaci a propojování lidí.

V roce 1995 Randy Conrad vybudoval první internetovou sociální síť classmates.com, která již tehdy měla dost společného se současnými sociálními sítěmi. Tyto webové stránky pomáhaly registrovaným uživatelům hledat a udržovat vztahy mezi spolužáky, studenty a jinými osobami. Dnes web čítá přibližně 40 milionů uživatelů, z nichž většina pochází hlavně ze Spojených států a Kanady [20] .

Koncepce classmates.com se ukázala jako úspěšná a proto ji následovaly světové giganty jako MySpace[1], FaceBook[2], Bebo[3] nebo LinkedIn[4].

[1] http://www.myspace.com/
[2] http://www.facebook.com/
[3] http://www.bebo.com/

Nejdůležitějším faktorem vývoje telekomunikačních sítí je možnost komunikace mezi osobami:

a) Elektronická pošta - tato forma elektronické komunikace vznikla jako první. Původně se emailová struktura používala pro výměnu zpráv mezi dvěma osobami, ale menší modifikace umožnila výměnu informace mezi skupinami lidí. Touto modifikací se staly mail-listy.
b) Telekonference nebo informační skupiny - telekonference se stala další etapou vývoje systému výměny informací. Jejích specialitou se stalo uchovávání zpráv, jejich třídění a poskytování přístupu k archivům.
c) Interaktivní chaty - rozvoj telekomunikací vedl k tomu, že čím dál více lidí začalo pracovat na internetu online, a proto je logické, že se objevily služby usnadňující takovou komunikaci. Specializovaná služba byla pojmenována Internet Relation Chat (IRC). V rámci této služby komunikace probíhá přes speciální internetové uzly - kanály.

Na samém počátku nebyla komunikace mezi uživateli v těchto službách hlavním cílem. Hlavní cílem bylo přesné zajišťování reálných problémů: šíření informace, diskuse, obchodní komunikace. Nicméně, časem se tyto technologie staly dostupné i běžným uživatelům z důvodu poklesu cen za připojení a hardware. Komunikace dostala více svobody a v rámci výše zmíněných služeb se formovaly komunity - skupiny lidí spojených společnými zájmy, pro které se komunikace s jinými členy stávala delší a intenzivnější než uvnitř těchto komunit. Často se online komunikace proměnila i v reálné seznámení. Taková online společenství měla rysy, které vyplývaly z jejich technické realizace:

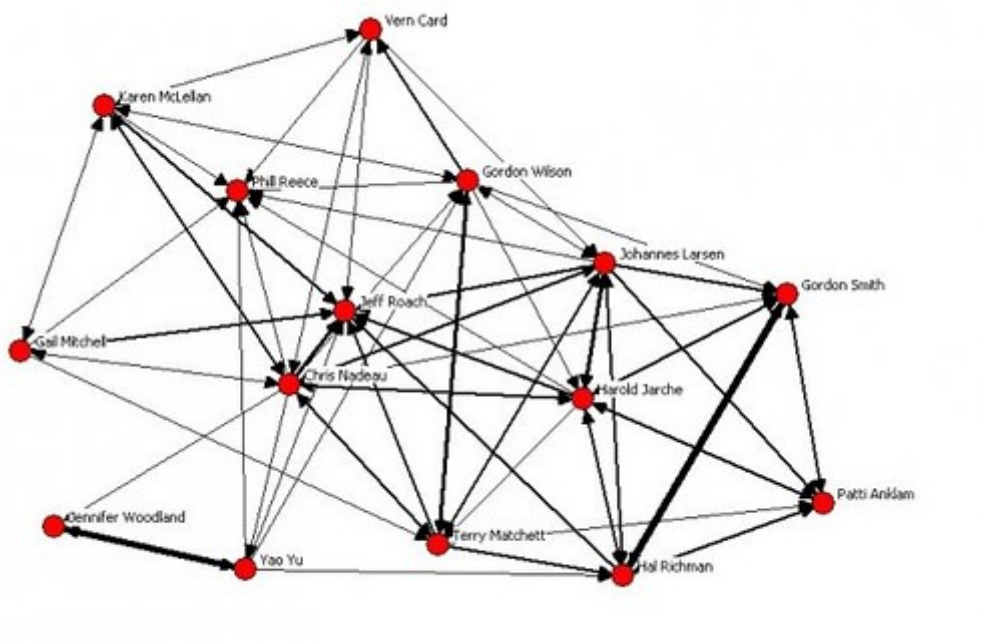
a) Pravidlem je identifikování uživatele v systému pomoci nějakého formálního jména, často zkráceného (nickname (angl.) - zdrobnělina, přezdívka). Logicky to umožnilo jednomu uživateli registrovat více přezdívek a tím pádem mít několik virtuálních postav.
b) Hlavním způsobem komunikace byla a zůstává výměna textových zpráv. Jelikož se ale pocity těžko sdělují pomoci textu, objevily se znaky pomáhající emocionálně zabarvit sdělení) - smajly (smile — úsměv).
c) Charakteristickou odlišností skupiny lidí, trvalé udržující nějaký vztah, je i shromažďování sociální historie vztahů. Tato historie částečně slouží k rozpoznání «cizí-svůj» a jedním z jejích projevů vztahů se stal svérázný styl komunikace, dialogových zkratek - žargonu.
d) Komunikace v podobných prostředích byla zavedena částečně i proto, že existuje již zmiňované zastoupení v podobě virtuální postavy, která nemusí nutně sdílet sociální status «majitele»: pohlaví, vzdělání atd.

[4] http://www.linkedin.com/

4. Analýza sociálních sítí

Analýza sociálních síti (*social network analysis - SNA*) je metoda/přístup, jejímž předmětem zájmu je rozbor sociálních vazeb jedinců [11] . Množinu těchto vazeb a účastníků tvořící sociální síť, kterou si lze představit jako graf. V rámci kterého jsou aktéři pojímáni jako uzly (*nodes*) a vazby, které je spojují jako hrany (*edges*). Prostřednictvím matematického vyjádření vlastností vazeb a pohledu na sociální sít jako na graf je jí možné podrobit analýze. Příklad grafového zobrazení sociální sítě je uveden na obrázku 4.1.

Tvar sociální sítě pomůže určovat užitečnost sítě k jeho jednotlivcům. Menší, těsnější sítě mohou být méně užitečné pro jejich členy než sítě s větším počtem volných spojení mezi jednotlivci v síti. Jinými slovy, skupina přátel, kteří dělají věci jen spolu navzájem, už mezi sebou sdílejí stejné znalosti a příležitosti. Skupina jednotlivců ve spojení s jinými sociálními světy má pravděpodobně přístup k širšímu rozsahu informací. Pro individuální úspěch bude lepší mít spojení k mnoha sítím, než pouze mnoho spojení uvnitř jediné sítě. Jednotlivci mohou podobně ovlivňovat nebo se chovat jako makléři uvnitř sociálních sítí, přemostěním dvou sítí, které nejsou přímo spojené.



Obrázek 4.1: Gráfova reprezentace sociální sítě

4.1 Sociální síťový rozbor

Sociální síťový rozbor (také nazývaný teorie sítě) se ukázal jako klíčová technika v moderní sociologii, antropologii, sociální psychologii, informatiky a organizačního studia [8] .

Teorie sítě se staví hlavně na rozboru grafu sítě pomocí různých technik známých z teorie grafu, analýzy časti grafu sítě, jeho uzlů a hran. Pro studium grafu sítě jsou důležité a známe takzvané indexy a centrality grafu. K analýze sociálních sítí, práce s indexy a centrality grafu sociální sítě se využívají specializované počítačové programy, mezi něž patří např. UCINET[5], Pajek[6], NetMiner[7], Gephi[8] atd.

Indexy pro sociální síťový rozbor:

Soudržnost

- Odkazuje se na míru jak moc jsou účastníci spojení k jiným skupinám účastníků. Skupiny jsou poznány jako "kliky", jestliže každý účastník je přímo svázán ke každému jinému účastníkovi nebo "sociálním kruhům", kde je menší hustota přímého kontaktu.

Hustota

- Individuální-hustota úrovně neboli globální-hustota úrovně je podíl skupin v síti vztažené k celkovému možnému množství.

Optická délka

- Vzdálenosti mezi páry uzlů v síti. Cesta průměru-délka je průměr těchto vzdáleností mezi všemi páry uzlů.

Radiality

- Míra toho jak jednotlivec působí do sítě, poskytuje nové informace a vliv

Dosah

- Míra toho jak nějaký člen sítě může dosáhnout propojení jiných členů sítě.

Strukturální rovnocennost

- Se odkazuje na rozsah uzlů, které mají obyčejný soubor vazeb k ostatním uzlům v systému. Uzly nemusí mít spojení s jinými uzly.

Strukturální díra

- Statické otvory, které mohou být strategicky zaplněny propojením jednoho nebo více spojení pro spojení s jiným uzlem. Tento pojem se ztotožňuje s myšlenkou pro sociální kapitál: "jestliže spojujete dvě osoby, které nejsou spojené, budete moct řídit jejich komunikaci".

Clustering coefficient (Koeficient sdružování)

- V teorii grafů koeficient sdružování je míra, do jaké se uzly v grafu shromažduji spolu ke skupině. Důkazy nasvědčují tomu, že ve většině sociálních sítí, uzly mají tendenci vytvářet sevřené skupiny vyznačující se relativně vysokou hustotou vazby. V reálném světě-sítě je toto

[5] http://www.analytictech.com/ucinet/
[6] http://vlado.fmf.uni-lj.si/pub/networks/pajek/
[7] http://www.netminer.com
[8] http://gephi.org/

nebezpečí zpravidla větší než průměr pravděpodobnosti rovnosti náhodně vytvořené mezi dvěma uzly.

- Existují dvě verze tohoto koeficientu: globální a lokální.
- Globální koeficient sdružování je založen na trojici uzlů. Triplet jsou tři uzly, které jsou spojeny buď dvěma (otevřené triplet) nebo třemi (uzavřené triplet) neorientovanými vazbami. Trojúhelník se skládá ze tří uzavřených trojčat, kdy jeden se soustředil na každém z uzlů. Koeficient globálního shlukování je počet uzavřených trojčat (nebo 3 trojúhelníky) na celkový počet trojic (oba otevřené a uzavřené).
- Místní koeficient sdružování z vrcholu v grafu kvantifikuje, jak blízko jeho sousedé mají být ke klice (úplný graf). Kliky v grafu je takový podgraf nějakého grafu, který je úplným grafem. Kliky se mohou překrývat. Výskyt kliky v grafu reprezentuje velkou soudružnost nějaké skupiny.
- Mezi další indexy sítového rozboru patří omezení, nákaza, integrace a jiné.

Avšak nejčastěji používané a nejzajímavější pro nás jsou indexy takzvané centrality. Více se budeme věnovat centralitám grafu pro případ sociální sítě v jedné z dalších kapitol.

4.2 Pomocné nástroje pro SNA

SNA software usnadňuje kvantitativní nebo kvalitativní analýza sociálních sítí, popisující vlastnosti sítě, a to buď prostřednictvím číselné, nebo vizuální reprezentace.

SNA softwary se obecně skládají, buď z balíčku založeného na grafickém uživatelském rozhraní (GUI), nebo z balíčku pro skriptovací nebo programovací jazyky. GUI balíčky jsou snadnější pro použití, zatímco skriptovací nástroje jsou silnější a rozšiřitelné. Široce používané a dobře zdokumentované jsou GUI balíčky obsahující UCINet pro statistickou analýzu sítí. Pajek, který je zdarma a pro které existuje rozsáhlá dokumentace, GUESS, ORA a Cytoscape jsou z mnoha možností, nejvíce rozšířené na těchto Gui-založených softwarových balíčcích. Ke GUI balíčkům zaměřeným na firemní zákazníky patří: Orgnet, který poskytuje školení o používání svého softwaru, a KXEN. Ostatní SNA platformy, jako jsou Idiro SNA Plus, byly vyvinuty speciálně pro určitá odvětví průmyslu, jako jsou telekomunikace a on-line hry, kde je třeba analyzovat masivní shromažďovaná data.

Mezi běžně používané a dobře zdokumentované skriptovací nástroje používané pro analýzu sítě patří statnet, igraph, který má balíčky pro výzkum a dále Python, NetworkX knihovna pro Python, a SNAP balíček pro sítě ve velkém měřítku Analýza v C++. Ačkoli je těžké se naučit tyto otevřené zdrojové balíčky, rostou mnohem rychleji v přepočtu na funkce a vlastnosti, než stávající software s rozsáhlou dokumentací a návody které jsou již k dispozici.

Vizuální reprezentace sociálních sítí jsou důležité, pro pochopení síťových dat a následné zpracované výsledky analýzy. Vizualizace často také usnadňuje kvalitativní interpretaci dat v síti. S ohledem na vizualizaci, analýzy sítě jsou nástroje zvyklé na změnu rozložení, barvy, velikosti a dalších vlastností zastoupených v síti.

Pro tuto práci byly použity dva Open Source nástroje a to Gephi a R.

Gephi je nástroj určený pro interaktivní vizualizaci, průzkum platformy pro všechny druhy sítí, komplexních systémů a dynamické a hierarchické grafy. Je to nástroj pro lidi, kteří mají prozkoumat a pochopit grafy. Uživatel může pracovat s reprezentací grafu, manipulovat s konstrukcí, tvarem, barvou a odhalovat skryté vlastnosti. Využívá 3D rendeerování pro zobrazení velké sítě v reálném čase a pro urychlení průzkumu. Flexibilní a více úkolová architektura přináší další nové možnosti pro práci s komplexními daty a hodnotnými vizuálními výsledky. Gephi umí nejen zobrazit graf, ale poskytuje možnost výpočtů a zobrazení základních statistik celého grafu nebo jeho komponent. Uživatelské rozhraní tohoto nástroje je příjemné a lehce pochopitelné pro jakéhokoliv uživatele, bez nutnosti předchozích zkušeností v práci s grafem nebo IT technologiemi.

Gephi má rozšířené spektrum vstupních a výstupních formátů: CSV, graphml, .dot, gml, tpl, net, aj.

V této práci byl použit Gelphi verze 7.0 beta, především pro zobrazování grafů, a pro vizualizaci mezi výsledky a jeho komponent.

R ("GNU S") - jazyk a prostředí pro statistické výpočty a grafiku. R je podobné systému S, který byl vyvinut v Bellových laboratořích John Chambers et al. Poskytuje širokou paletu statistických a grafických metod (lineární a nelineární modelování, statistické testy, analýzy časových řad, klasifikace, shlukování, aj.).

R obsahuje několik balíčků pro relevantní analýzu sociální sítě:

- igraph - je obecný balíček pro síťovou analýzu;
- sna - provádí sociometrickou analýzu sítí nebo manipulaci a zobrazení síťových objektů;
- tnet - provádí analýzu vážených sítí, dvou-režimů sítě, a podélných sítí;
- ergm - nástroj exponenciální náhodné grafické modely pro sítě;
- latentnetmá - funkce pro latentní polohy sítí a klastrů modelů;
- degreenet - poskytuje nástroje pro statistické modelování stupňů rozvodů sítí
- networksis - poskytuje nástroje pro simulaci bipartitního grafu s pevnou poznámkou.

Většina z těchto balíčků jsou součástí statnet apartmá, které je možné získat prostřednictvím statnet meta-balíček.

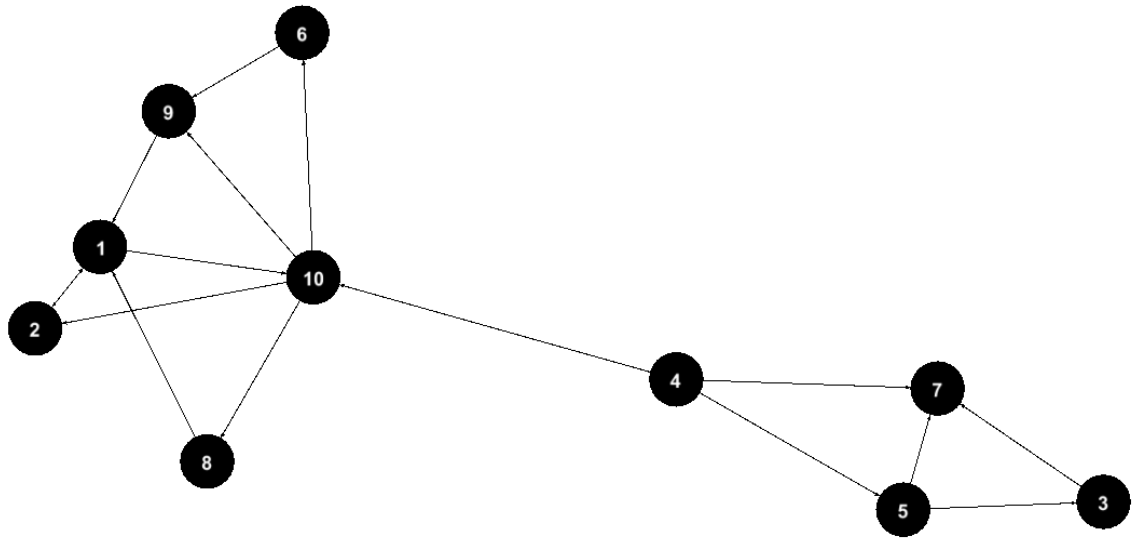
Uživatelské rozhraní R se razantně liší od Gephi. Práce v R je obtížnější a výsledky nejsou tak přehledné, avšak pomocí existující rozsáhlé dokumentace je uživatel schopen pracovat i s tímto nástrojem.

R byl použit v této práci hlavně pro ladění algoritmů výpočtu centralit grafu sítě.

4.3 Centrality

V rámci teorie grafů a síťové analýzy, existují různé míry centralit vrcholů v grafu, které určují relativní důležitost vrcholu v grafu (například, jak důležitá je osoba v rámci sociální sítě, nebo dle teorie z prostoru syntaxe atd.) [12] .

Vyčleňme si několik druhů centralit grafu, s kterými se budeme zabývat dále a které by mohly být zajímavé nebo nejpřínosnější pro analýzu sociální sítě: Betweeness centrality, Closeness centrality, Degree centrality, Clustering coefficient, Eigenvector centrality [12] . Pokusíme se vysvětlit každou z centralit na menším grafu sociální sítě, který se skládá z 10 uzlů a 16 hran. Graf je zobrazen na obrázku 4.3.1. Tento graf reprezentuje sociální sít členů školního týmu, kteří spolu pracují na semestrálním projektu. Ke každému uzlu/studentovi přiřadíme id. Výpočty statistik v této fázi budeme provádět pomocí aplikace R a zobrazení výsledků pomocí aplikace Gephi 0.7, jak jsme se již zmínili v předchozí kapitole.



Obrázek 4.3.1: Graf sociální sítě tymu studentů.

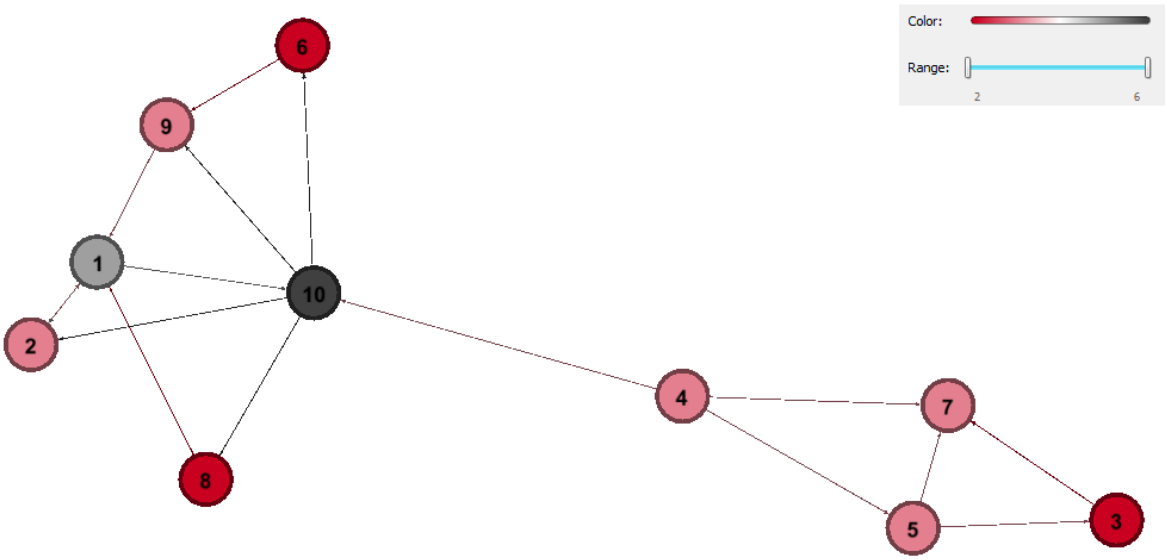
Degree centrality

První a nejjednodušší centralita je Degree centrality. Tato centralita je definována jako počet odkazů dopadajících na uzel (tj. počet vazeb, které má uzel) [12] . Degree je často označován z hlediska bezpečnosti, jako potencionální hrozba uzlů, rizik, která protékají sítí. (např. virus nebo nějaké informace). Pokud je směrována sítí (což znamená, že vazby jsou orientované), pak se obvykle definují dvě různé míry opatření a to indegree a outdegree. Indegree je součet počtu vazeb, které směřují do uzlu. Outdegree je počet vazeb, které uzel směřuje k jiným.

Pro graf sociální sítě degree centralita říká, jak moc komunikativní je účastník sítě a jak velkou popularitu má mezi jinými účastníky. Tyto výsledky a vyhodnocení si zobrazíme na ukázkovém grafu studentů. Vypočteme degree centralitu pro každý uzel. Výsledky jsme zobrazily tabulkou hodnot centralit pro každý z uzlů a obrázkem (viz. Tabulka 4.3.1 a Obrázek 4.3.2), kde pomoci barev

jsou popsány velikosti hodnoty degree. Výsledky se zobrazují v tabulce hodnot centralit pro každý z uzlů a obrázkem, kde jsou pomocí barev popsány velikosti hodnoty degree. Z obrázku je vidět, že se hodnota degree centrality v této síti pochybuje od 2 po 6. Když vezmeme v úvahu celkový počet účastníků v síti a průměrnou hodnotu degree centrality uzlů, jsme schopni říct, že tato skupina lidí je mezi sebou dost propojená a nejsou v ní osamocení jedinci. Taky můžeme tvrdit, že úroveň komunikativnosti mezi účastníky v síti je skoro stejná s výjimkou účastníka 10.

Hodnota degree centrality účastníka 10 se razantně liší od degree jiných účastníků. Z čeho jsme schopni posoudit, že tento účastník je pravděpodobně nejpopulárnější, neboli nejkomunikativnější účastník v této síti.



Obrázek 4.3.2: Degree centrality

ID uzlu	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Degree	4	2	2	3	3	2	3	2	3	6

Tabulka 4.3.1: Degree centrality

Betweeness centrality

Betweeness je ústřední míra blízkostí vrcholu v grafu. Vrcholy, které se vyskytují na mnoha cestách mezi ostatními vrcholy, mají vyšší betweeness než ty, které se nevyskytují na mnoha cestách mezi ostatními vrcholy [12] . Od ostatních centralit se liší tím, že se z něho neměří, jaká je konektivita vrcholu, ale měří se jak vrchol „zapadne“ mezi ostatní.

Pro graf $G := (V, E)$ s n vrcholů, Betweenness $C_B(v)$ pro vrchol v se vypočte následovně:

1.1 Pro každou dvojici vrcholů (s, t) se vypočítají všechny nejkratší cesty mezi nimi.

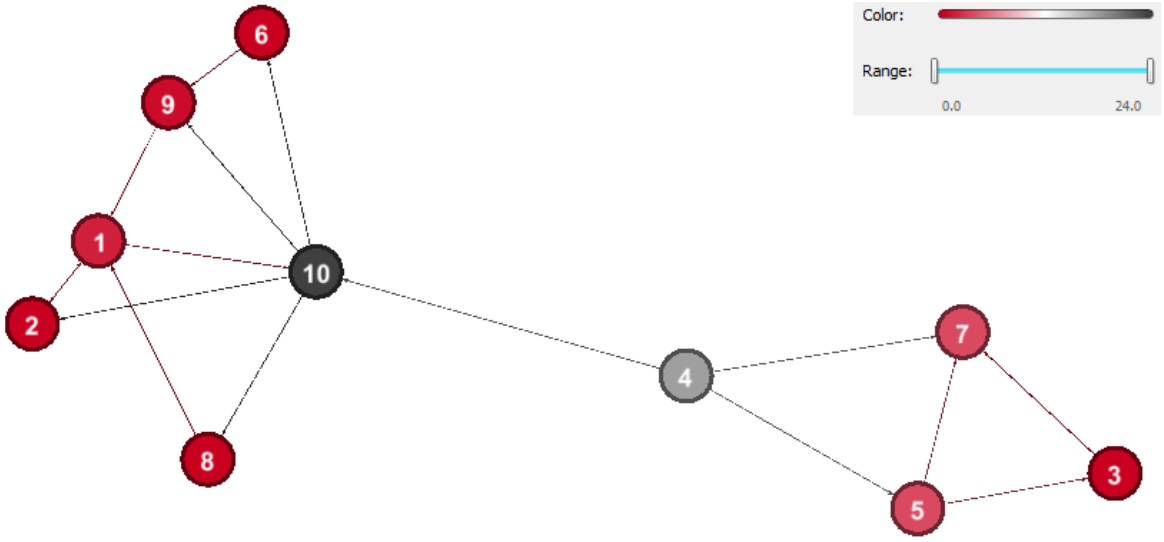
1.2 Pro každou dvojici vrcholů (s, t) se urči podíl nejkratší cesty, která prochází vrcholem v.

1.3 Vypočítá se součet těchto zlomků nad všemi dvojicemi vrcholů (s, t). Jinak:

$$C_B(v) = \sum_{s \neq v \neq t \epsilon V} \left(\frac{\sigma_{st}(v)}{\sigma_{st}} \right)$$

kde
σ_{st} je počet nejkratších cest z s do t,
$\sigma_{st}(v)$ je počet nejkratších cesty z s do t, které projdou přes vrchol v.

Vypočítali jsme betweeness pro každý uzel naší pokusné sítě. V tabulce 4.3.2 zobrazujeme hodnoty centralit pro každý z uzlů a obrázek 4.3.3, kde jsou pomocí barev popsány velikosti hodnoty betweeness. Z obrázku je vidět, že se hodnota betweeness v této síti pochybuje od 0, až po hodnotu 24. Největší hodnotu betweeness má uživatel s id 10. Nejmenší hodnoty betweeness mají uživatelé 1, 2, 8, 9, 6 a 3.



Obrázek 4.3.3: Betweeness centrality

ID uzlu	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Betweeness	0	0	0	18	1	0	6	0	0	26

Tabulka 4.3.2: Betweeness centrality

Pojem betweeness v kontextu sociální sítě, můžeme pochopit jako míru důležitosti uzlu pro spojeni jiných účastníků mezi sebou. To znamená, že čím vetší je hodnota betweeness uzlu, tím podstatnější je tento uzel pro síť.

Vidíme, že z provedeného experimentu je pravděpodobně nejdůležitější, pro propojování jiných uzlů mezi sebou v síti, uživatel s id 10. Naopak uzly 1, 2, 8, 9, 6 a 3 mají slabší (menší) míru důležitosti [12] .

Closeness centrality

Blízkost centrálnosti je míra, jak je jednotlivec blízko ode všech jiných jednotlivců v síti (přímo nebo nepřímo) [10] .

V teorii grafů je blízkost míra centrálnosti vrcholu v grafu. Vrcholy, které jsou 'blízko' k dalším vrcholům (tedy ty, které mají krátké vzdálenosti přímou čarou na další vrcholy v grafu), mají vyšší těsnost a vyšší hodnotu closeness centrality.

Při výpočtu closeness centrality se používají nejen sousedi uzlu, ale i sousedi sousedů (uzly, které nejsou přímo spojené s tímto uzlem). Nepřímo spojené uzly obdrží nižší hodnotu, protože intenzita jejich vztahu, nebo jejich vliv je menší.

Closeness centrality pro sociální sít můžeme pochopit tak, že je-li jeden účastník, hodně blízko od jiných účastníků (má vetší hodnotu closeness centrality) v síti a v návaznosti na to má tento účastník možnost rychlejší komunikace se všemi ostatními uzly v síti. Obecně řečeno, closeness centrality popisuje míru propojeností uzlů v síti. Uzel, který má velkou hodnotu closeness centrality (je připojen krátkými cestami do jiných uzlů), kdy ty mohou být interpretovány jako relativně autonomní uzly.

Pro příklad významnosti closeness centrality jsme ji vypočetli pro každý uzel naší ukázkové sítě studentů. Výsledky jsme zobrazili v tabulce 4.3.3. Z analýzy obrázku plyne, že nejrychlejší spojení z jinými uzly v síti (nebo uzly sítě) mají studenti 10 a 4 (tyto uzly májí nejmenší hodnoty closeness centrality).

Naopak nejmenší hodnotu této centrality má uzel 3. To znamená, že má nejdelší cesty k ostatním uzlům sítě a je více osamocený v rámci sítě než ostatní.

ID uzlu	Closeness
1	0.05555556
2	0.05000000
3	0.03703704
4	0.06666667
5	0.05000000
6	0.05000000
7	0.05000000
8	0.05000000
9	0.05263158
10	0.07692308

Tabulka 4.3.3: Closeness centrality

Eigenvector centrality

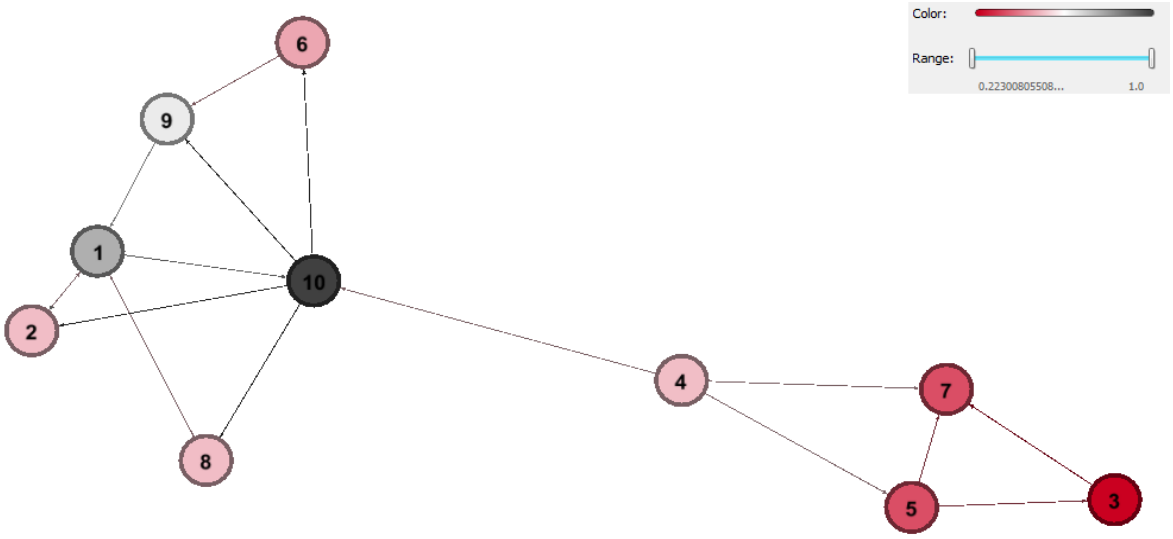
Eigenvector centralita je míra důležitosti uzlu v síti. Tato centralita je rozšířením degree centrality a zohleduje to, jak významné jsou sousední vrcholy vrcholu, pro něž centralitu určujeme. Každému vrcholu přiradí hodnotu úměrnou součtu hodnot sousedních vrcholů

$$x_i^{'} = \sum_j A_{ij}\, x_j, \qquad nebo\ x^{'} = Ax$$

Kde A_{ij} je prvek matice sousednosti, respektive x je vektor prvků x_i [13] .

Tím pádem pro graf sociální sítě můžeme tvrdit, že čím větší je hodnota eigenvector centrality, tím důležitější je uzel, což může byt užitečné při analýze komunit a hledaní roli účastníků.

Příklad výsledků výpočtů eigenvector centrality na testovacím grafu sociální sítě ukazujeme na obrázku 4.3.4 a tabulce 4.3.4. Po analýze obrázku a tabulky hodnot centrality můžeme ohodnotit, že nejdůležitější účastník v této sítě je student s id 10.



Obrázek 4.3.4: Eigenvector centrality

ID uzlu	Eigenvector
1	0.7930332
2	0.5239215
3	0.1362223
4	0.4284207
5	0.2330990
6	0.4868551
7	0.2330990
8	0.5239215
9	0.6661798
10	1.0000000

Tabulka 4.3.4: Eigenvector centrality

4.4 Matematický pohled na SNA

V současné době je analýza sociálních sítí značně matematizovaná, data jsou zpracovávána počítačovými programy. Nejsložitější analýzy se zabývají formalizovanými dynamickými modely procesu vzniku sociálních sítí, a to i na bázi simulovaných dat. K analýze sociálních sítí slouží relační data (tj. kontakty, vazby a spojení dvou jedinců) uspořádaná do sociogramů a matic. V prvních sociometrických výzkumech výběru přátel používal Moreno (1934) dotazníky a překročil tak omezení výzkumu skupin pozorováním. Základní sociometrická matice v sobě obsahuje zjištění, s kým je (nebo se považuje být) každý jednotlivý z

n aktérů spojen nebo ne-spojen vazbou. Teorie grafů, která používá algoritmy, umožňuje z výpočtů údajů v maticích stanovit obecné charakteristiky sociální sítě (konexita, vzdálenost), soubory homologických pozic (kliky, třídy ve stejné strukturální úrovni) nebo odhalit zaznamenání hodné body v síti (centralitu, mosty) [18] .

Diagramy [18] sociálních sítí, které znázorňují vzory spojení mezi body, mohou být popsány dle Scotta [1991] těmito základními pojmy:

- Orientovaný graf znázorňuje šipkou orientovaná spojení od jednoho aktéra (zde bodu v grafu) k druhému a směr šipky indikuje orientaci vztahu. Dva body spojené vazbou jsou považovány za přilehlé a tvoří sobě navzájem rozličně velké sousedství.
- Body mohou být spojeny různým způsobem, i zprostředkovaně. Sekvence vazeb – spojnic, které vedou a zprostředkovávají vztah jednoho bodu s druhým, se nazývá cesta a délka cesty se měří počtem vazeb, které tvoří spoj.
- K celkovému popisu studovaného grafu slouží proměnná hustota a centralita nebo centralizace. Pojem „hvězda“ pochází se sociometrie, je to osoba ve skupině nejvíce oblíbená, jenž je středem pozornosti. Centrální bod může mít lokální nebo globální charakter. Bod je centrální lokálně, jestliže je spojen větším počtem vztahů s jinými body v jeho nejbližším okolí. Bod centrálně globální je takový, který má strategickou pozici v celkové struktuře sítě a zároveň leží v krátké vzdálenosti od ostatních bodů. Zaměnit na centrálně globální body s více centrální povahou ve srovnání s těmi, se kterými jsou spojeny, jsou vrcholy; mosty jsou pak centrální body, které spojují dva nebo více vrcholy.

Strukturovat sociální síť lze dle klik nebo kohezních podskupin (např. klastrů, komponent, kruhů). Komponenta je izolovaná skupina i zprostředkovaně spojených bodů v síti. Klika (neformální uskupení lidí, mezi nimiž byly ustaveny skupinové normy chování a kteří spolu soucítí) je podmnožina bodů, kde každé dva body jsou přímo spojeny, jsou tedy přilehlé. Vzájemně se překrývající kliky (jejichž hranice lze vymezit snowballingem) mohou být agregovány do kruhů. Klastr je husté uskupení sobě si podobných bodů, které nemá jasné hranice. Na principu podobnosti, lze pak v dendrogramech seskupovat klastry do stejných hladin. Body (aktéři) se třídí do aglomerací podle toho, s kým sousedí s přihlédnutím k tomu, zda jsou nositeli předem definovaných atributů [11]

Vedle popisu vzorů uspořádání přímých a nepřímých vazeb lze i detailněji analyzovat vztahy mezi aktéry ve stejných pozicích. Problém srovnatelnosti sociálních pozic těch, kteří jsou seskupeni do klik nebo klastrů, se řeší konceptem strukturální ekvivalence.

4.5 Analýza Log souborů

Moderní aplikace jako informační, podnikové a e-commerce systémy stejně jako monitorovací, webové a jiné aplikace generuji obrovské množství dat. Této data jsou různých druhů. Mohou to byt základní textové informace ve formátu čistého textu jako logy nebo informace předané přes HTML nebo XML formáty, a

nebo polo-strukturované multimediální data (audio, video soubory atd.). Nejčastěji podobné sbírky dat se udržuji v databázích, datových skladištích nebo jednoduše v datových logových souborech [6] .

Analýza logových souborů nabývá nárůstu pozornosti z každé oblasti lidských aktivit. Tento obor však není zajímavý pro průzkumní oblast, ani pro oblast vývoje softwarů v komerční sféře. Obsahuje v sobě discipliny cílené na analyzování datových zdrojů pro získání hodnotné informace, často přestavěné jako vědomostí. Získané informace (vědomosti) jsou často používané pro management, udržovaní, zlepšeni systémů, které byli tvořené těmito informacemi, nebo pro jiné cíle, jako objevovaní struktury sítě nebo struktury sociálních sítí. Jestli záznam logu obsahuje informace o personách, které popisují jednotlivé aktivity, říká o potenciální možnosti využití těchto dat pro analýzu sociálních sítí.

Rozbor logových souborů – proces získaní informace orientovaný na analýzu záznamů generovaných počítačem. Rozbor logů se provádí pro různé účely, například systémová bezpečnost, regulovaní procesů, odchytávání systémových chyb, analýza sociálních sítí atd.

Logový soubor – jednoduchý text, generovaný zařízením, softwarem, aplikace, nebo samotným systémem. Log obsahuje hlavně zprávy, které poskytuji informace o aktivitách. Typicky log soubor se skládá z informace o aktivitě anebo události, většinou nese v sobě časovou značku, informace o odesilatelích a jiné data. Odesilatel může byt zaznamenán jako persona nebo zařízeni, je to souvisle s typem log souboru.

Například typicky webový log je uložen webovým serverem jako záznam aktivity návštěvníků webových stránek. Tento logový soubor má standardní formát a obsahuje následující informace: IP adresu klienta, který přistupoval na webové stránky, uživatelské jméno, datum a čas požadavku, zdroj požadavku, velikost v bajtech dat, které byly vrácené klientu a URL, která odkazuje na klientsky zdroj. Níže uvádíme menší příklad možného obsahu log soboru, kde každý řádek reprezentuje určitou událost a popisuje její vlastnosti: datum a čas, id a typ události, id a typ objektu, id a jméno uživatele, který tuto událost prováděl.

```
2007-03-10 17:17:55;337777;CreateEvent;11;D1;2;Bob;
2007-03-11 17:19:15;333481;CreateEvent;13;D3;1;Alice;
2007-03-16 09:13:22;335481;ReadEvent;13;D3;2;Bob;
2007-03-17 12:17:56;385481;ReviseEvent;13;D3;2;Bob;
2007-03-17 13:17:45;337431;ReadEvent;12;D2;2;Bob;
2007-03-17 14:19:35;332581;ReviseEvent;12;D2;1;Alice;
2007-03-17 16:10:25;346541;ReadEvent;12;D2;1;Alice;
2007-03-18 13:25:15;312431;ReviseEvent;11;D1;1;Alice;
```

5. Sociální sítě v oblasti spolupráce vývojářů.

Jak již bylo zmíněno dříve, sociální síť se tvoří na základě zájmů, rodinných vazeb nebo z jiných důvodů. Důvody k vzniku sociálních síti a navazování komunikace mezi lidmi v naší době jsou různorodé: studium, seznamování, společná turistika a cestovaní, práce, hry a výjimkou není ani programování. Hodně programátorů hledá na internetu zajímavé nápady, nebo pomoc při realizaci nápadů vlastních. V naší době už není ani novinkou spolupráce on-line, která je provozována na delší vzdálenosti. Stále aktuální je však problém získání financování a podpory při realizace zajímavých softwarových nápadů. A naopak, i při existenci finančních zdrojů není vždy jednoduché najít perspektivní a zajímavý námět a zároveň skupinu schopných lidí, která by realizovala konkrétní námět v rozumných termínech a vysoké kvalitě. Těmto problémům se samozřejmě snaží napomáhat sociální sítě a internet.

OSS (Open Source Software) – toto je klasický příklad dynamické sociální sítě a zároveň prototyp komplexně rozvijících se sítí v internetu. Developeři spolupracují přes internet a tím tvoří sociální sít. Formace sociální sítě vzniká ve chvíli, kdy se vývojář připojí do projektu a komunikuje s jinými vývojáři. Mission[9], CodePlex[10], SourceForse[11] – jsou příklady světově známých sítí OSS.

OSS sociální sít je tvořena dvěma entitami – vývojáři a projekty. Pomocí tohoto můžeme sociální sít z bipartitního grafu transformovat do unipartitního grafu: sítě vývojářů a sítě projektů, jejichž uzly jsou navzájem propojené mezi sebou. Transformace do unipartitního grafu sociální sítě nese s sebou řadu výhod, které si ukážeme dále. Spojení mezi dvěma projekty indikuje to, že oba projekty mají jednoho nebo více společných vývojářů. To znamená, že se již tento vývojář zúčastnil více projektů. Komunikace mezi vývojáři v rámci OSS většinou probíhá pomoci mailu, na fóru, v chatu atd.

V dnešní době existují nástroje pro komunikace vývojářů jak komerční, tak i nekomerční. Většina nekomerčních portálu postrádá na přehlednosti uživatelského rozhraní a možnosti vyhledávaní jak uvnitř portálu, tak i dohledaní informací umístěné zvnějšku na portále. Není vždy jednoduché najít to, co hledáte, buď informace, nebo lidi. Komerční řešení často nabízí reklamy a různé benefity pro vývojáře. Jsou lehce dohledatelné na webu a mají dokonalejší uživatelské rozhraní.

5.1 Aspekty týmové práce

Potřeba být členem (lidského) společenství, někam patřit, je jednou ze základních hnacích sil našeho života, která nás vede k seznamování s novými lidmi, k tomu, že se snažíme dostávat do různých skupin. A tato potřeba spolu s potřebou uznání nás v těchto skupinách vede k postavení se do určitých rolí a k vykonávání různých činností, více či méně ku prospěchu cele skupiny, týmu.

[9] http://www.opensource.org/
[10] http://www.codeplex.com/
[11] http://sourceforge.net/

Slovo tým znamená v původním staroanglickém termínu "spřežení, potah". V přeneseném smyslu týmem rozumíme sportovní mužstvo. V obou případech působí jejich složky (koně či hráči) jako celek a plní společný cíl.
Slovo tým naznačuje, že tu jde o přesah jednotlivce. Že tým je víc než jedinec. Pro všechny případy práce existují tři hlavní druhy týmů, které si můžeme přiblížit na následujících příkladech:

První typ týmu lze přirovnat k baseballovému týmu, každý hráč má své místo, které nikdy neopouští, nebo k týmu štafety na 4 x100 m, ve které má každý atlet přidělen svůj pevný úsek, na kterém běží úplně sám bez pomoci ostatních a který nesmí dokonce ani přeběhnout.

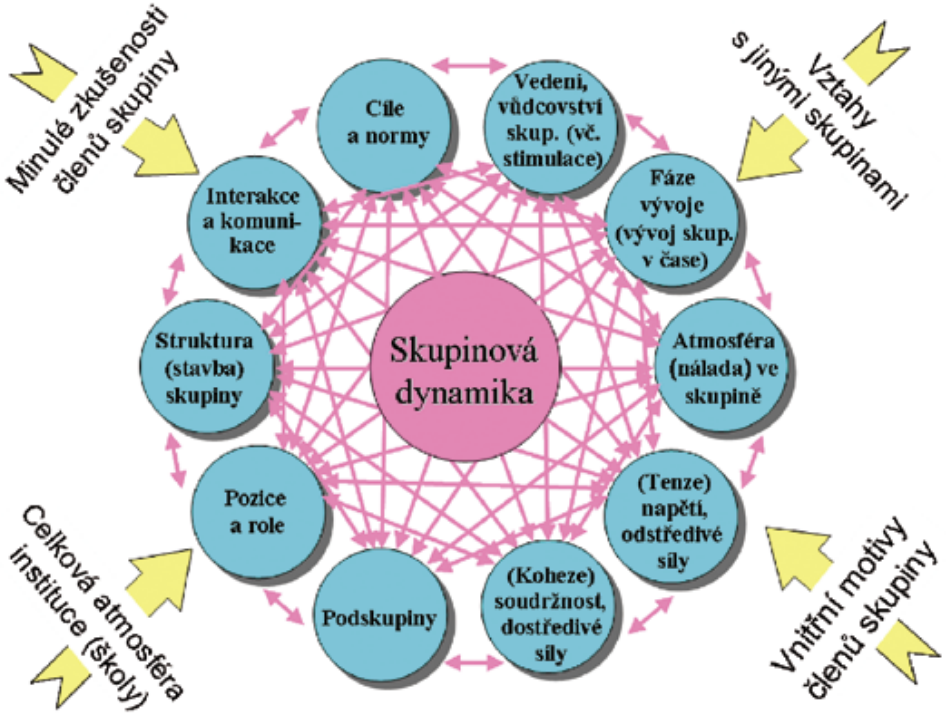
Druhým typem týmu je fotbalový nebo hokejový tým. Také v tomto týmu mají všichni členové týmu pevně určené místo. Ale úroveň součinnosti spolupráce je nepochybně vyšší než v předchozím typu týmu.

Třetí typ týmu je tým typu plážového volejbalu, jehož členové se "vykrývají", přizpůsobují se přednostem a slabinám svých spoluhráčů a oba umí všechno potřebné z abecedy plážového volejbalu. Dá se říct, že tento typ týmů je ze všech nejsilnější.

Skutečná týmová práce znamená především souhru, součinnost jednotlivců. Specifika týmové práce spočívá v tom, jak se této součinnosti dosahuje.

Týmová práce je efektivní forma organizace lidské práce, která má vícedimenzionální charakter, probíhá v trvalém rozvoji pracovních vztahů členů týmu, kteří mají určité pracovní role, nebo si je sami rozdělují a mění dle vlastní volby. Týmová práce je založena na principu oboustranného a otevřeného toku informací [13] .

Týmy, které lidé vytvářejí, jsou různě veliké, vznikají z různých důvodů, fungují na různých principech. V každé skupině bez rozdílu však
existují určité síly, určité procesy, které mají vliv na skupinu i na její členy. V každé skupině existuje a působí soubor nejrůznějších hybných sil a protisil. Tomuto souboru sil a protisil, jež ovlivňují skupinu a skupinové dění, se říká skupinová dynamika. Skupinová dynamika je složena z mnoha prvků, které se navzájem ovlivňují a vyvíjejí. Zásah do jednoho prvku ovlivní všechny ostatní a logicky i naopak. Na skupinovou dynamiku navíc působí i nejrůznější vnější síly a faktory. Na níže uvedeném obrázku jsou graficky znázorněny základní prvky skupinové dynamiky a jejich vztahy (dle PhDr. Zdenko Matuly) [13] .



Obrázek 5.1.1: Schémata základních prvků skupinové dynamiky a jejich vztahů [13] .

Stručný pohled na jednotlivé prvky skupinové dynamiky [13] :

Cíle a normy – v každé skupině, jsou-li její členové alespoň chvilku pospolu, se postupně vytvoří jisté cíle (kam /společně/ směřujeme). Individuální cíle jednotlivých členů se mohou lišit. Členové skupiny mohou vytvářet koalice a kliky (podskupiny) za účelem prosazení svých cílů ve skupině. O cílech se diskutuje, mohou kvůli nim a jejich rozdílnosti vznikat konflikty. Posléze vznikají nejrůznější pravidla a postupy (tj. normy), které by měly skupině pomoci dosáhnout cílů.

Vedení a řízení, motivace a stimulace – jsou-li ve skupině vytyčeny cíle a stanoveny normy, které mají zajistit dosažení cílů, pak vedení (a řízení) je způsob, jak se pracuje s normami a s procesy ve skupině vůbec, tak aby bylo cílů dosaženo. Termín „vedení“ v našem pojetí vyjadřuje více svobody a spoluúčasti pro ty, kteří jsou vedeni. Pojem „řízení“ naopak (stejně jako např. řízení auta) vyjadřuje, že ti, kdož jsou řízeni, pouze vykonávají rozkazy, které jim dává „řidič“, tedy ten, kdo řídí. Aby členové skupiny měli zájem o dosahování cílů a dodržování norem skupiny, je vhodné vytvořit určité stimulační prostředí, které se snaží oslovit individuální motivy členů skupiny
tak, aby dodržovali normy a snažili se o dosažení skupinových cílů.

Interakce a komunikace – nejvýraznější (viditelná) část vzájemného působení členů skupiny probíhá v oblasti interakce (vzájemné ovlivňování členů skupiny v co nejširším slova smyslu) a komunikace (verbální a neverbální). Způsob komunikace pak výrazně ovlivňuje zejména další oblasti skupinové dynamiky (např. atmosféru, způsoby vedení skupiny, jak skupina pracuje s cíli a normami, poměr dostředivých a odstředivých sil).

Podskupiny – dříve či později vznikají v každé (větší) skupině. Přičemž je důležité, do jaké míry se členové podskupin ztotožňují s normami a cíli celé velké (mateřské) skupiny.

Struktura, stavba skupiny – je tvořena systémem pozic a rolí v každé skupině. Takřka ve všech skupinách existují formální a neformální struktury. Tyto dvě struktury se mohou více méně překrývat či lišit.

Pozice a role – pozice určuje místo toho kterého člena skupiny ve skupině, jeho zařazení, „umístění“ (ve struktuře). Pozice určuje status člena skupiny (soubor jeho práv /výhod/ a povinností), určuje rozsah jeho moci (co smí a co nesmí). Role je proti tomu charakterizována souborem chování, která skupina očekává, že bude „hrát“ ten který její člen na dané pozici.

Fáze vývoje skupiny, historie skupiny – každá skupina se vyvíjí v čase, prochází určitými vývojovými stádii.

Atmosféra ve skupině – skupiny se liší atmosférou, tedy jakousi „náladou“ skupiny. Atmosféru vytváří členové skupiny ve svých každodenních interakcích. Atmosféra výrazně ovlivňuje „kulturu organizace“ („firemní či školní kulturu“), ve které je skupina začleněna (viz např. dnes hojně používaný pojem „skryté kurikulum“).

Soudržnost (koheze) a napětí (tenze) neboli dostředivé a odstředivé síly – mezi soudržnost či dostředivé síly řadíme všechny síly a mechanismy, které způsobují, že jednotliví členové skupiny chtějí do skupiny patřit, cítí se být členy skupiny. Napětí (tenze) či odstředivé síly jsou naproti tomu síly, které rozleptávají skupinu a způsobují, že (některým/všem) členům skupiny ve skupině není dobře, nechtějí do skupiny patřit. Přetlak a výron velkého množství odstředivých sil může v krajním případě způsobit i rozpad skupiny.

5.2 Metody analýzy týmu v rámci SN

Ve 4 kapitole jsme se zabývali analýzou sociálních síti jako celku a analýzou jednotlivých uzlů. V této kapitole se pokusíme ukázat, že pro nás může být důležitá i samostatná analýza části sociálních sítí, a to hlavně pro výzkum vlastností skupin účastníků sociálních sítí.

Pro realizaci určitých projektů na internetu se tvoří skupiny vývojářů, které spolupracují a podílejí se na projektech v průběhu jejich vývoje. Skupiny se tvoří buď dle vlastního zájmu jednotlivých účastníků, nebo je tvoří lídr/iniciátor projektu. Složení a porozumění uvnitř týmu je velmi důležité pro celkový úspěch a kvalitu výsledného díla v jakékoliv práci, zvláště v pracích typu on-line, kde se jednotliví účastníci nemusejí a často ani nemají možnost se vídat a znát. Je zřejmé, že měření výkonnosti týmu je tvořeno přes internet pomocí metod, které jsou v současné době velmi oblíbené, avšak tato oblast se jen začíná zkoumat.

Pro nalezení efektivních metod analýzy ohodnocení, týmů, jednotlivých účastníků byly podrobeny zkoumání již existující metody analýzy v rámci tymů nebo sítě spolupracujících navzájem mezi lidmi.

Některé metody analýzy týmu se opírají o analýzu dotazníků, které jsou povinné vyplnit členy týmu. Dotazníky se většinou vypracovávají v různých časových odstupech při běhu jednoho projektu. Z analýzy dotazníků jsme schopni zjistit variace týmové struktury během jednotlivých fází projektu, rozpoznat vztah mezi strukturou týmu a celkovou výkonností, také vyhodnotit lidery, nebo šikovné dvojce, podskupiny týmu, pasivní / aktivní účastníky atd. Následně se tyto výsledky analýzy dotazníků porovnávají se

strukturou sociogramu. Avšak s tohoto porovnávání nejsme schopni odvodit tvrzení, že ze samotné analýzy sociogramu dostaneme stejné výsledky. Bohužel metoda dotazníků nelze uplatnit všude, kde bychom potřebovali analyzovat práci v týmu.

Jiné metody se zaměřují na analýzu jedinců a následné sjednocení účastníků s podobnými vlastnostmi. Výpočet vlastností, každého uzlu v síti se dělá pro lepší pochopení polohy členů týmu v sítových vazbách, jejich organizace dle služebního věku, projektové role nebo geografického rozložení. Pod pojmem vlastnosti uzlu v kontextu této metody analýzy týmu se rozumí:

- výpočet stupně vzdálenosti (počet linek směrovaných k vybranému uzlu),
- výpočet indexů hustoty (nepočítá se spojení uzlu z jejich sousedy, ale se vzdálenými uzly),
- relaci typu "mezi" (hledá uzly zprostředkovatele / konektory. To znamená uzly, které slouží mezi dvojicemi uzlu jako spoji. To znamená, že jiná možnost navazovaní komunikace neexistuje.)
- a jiných vzorků pro uzly grafu sítě.

Jednou z důležitých součástí této analýzy je výpočet centrálnosti uzlu v síti, pro vypočet kterého musí existovat nadefinovaný koeficient míry, pro porovnání jednotlivých uzlů. Tato metoda analýzy může být užitečná při analyzování menších spolků lidí. Odhaleni společně spolupracujících skupin nebo podskupin pro porovnávání a ohodnocení jednotlivých účastníků. V rámci vetší komunity, však nejsme schopni potvrdit jednoznačnost výsledků této analýzy.

Příkladem jedné z nejzajímavějších metod analýzy týmů je analýza dynamické sociální sítě SourceForge.net. O základech této metody bylo zmíněno již v minulé kapitole, na kterou se teď podíváme důkladněji.

Analýza dynamické sítě SourceForge.net byla založena na transformování bipartitního grafu sítě do unipartitních grafů: síť vývojářů a sít projektů, uzly kterých jsou navíc propojené mezi sebou. Spojení mezi dvěma projekty v síti projektů indikuje to, že oba projekty mají jednoho nebo více společných vývojářů. Spojení mezi vývojáři v rámci sítě vývojářů indikuje to, že se oba vývojáři podílejí na jednom nebo více společných projektech [7] [8] .

Dle rozlehlosti a velikosti původní sítě můžeme pro rozdělení do unipartitních grafů přidat váhový koeficient, pomoci kterého dokážeme korektně vynechat méně důležitá spojení mezi uzly ve výsledných grafech. Tím samým dostaneme efektivní nastroj k rozčlenění a rozděleni sítě pro další využití. Je zřejmě, že rozdělení sociální sítě na dvě entity – vývojáři a projekty, nám poskytují možnost analýzy nejen jednotlivých účastníků projektu, ale i projektů celých, což zahrnuje hodnocení jejich úspěšnosti, porovnávání atd. Zároveň jsme schopni podle těchto projektů rozdělit celou sociální síť na menší části. A rozdělení sociální sítě dokáže razantně oblehčit pracnost a časovou náročnost analýzy.

Celkově SNA se považuje za silný nástroj pro odhalení vnitřní struktury týmu a jejích vlastností. Dokonce můžeme tvrdit, že pomocí studia vztahu mezi sociogramem částí sociálních sítí, která se týkala určitého projektu, a výsledné výkonnosti týmu, můžeme odvodit následující informace [4] :

- Lídry týmu
- Šikovné dvojice/podskupiny týmu
- Izolované skupinky nebo jednotlivce
- Populární/nepopulární účastníky
- Aktivní/pasivní účastníky, atd.

6. Příprava vstupních dat[12]

Pro vytvoření sociální sítě a experimentů nad ní byla použita reálná data z webu CodePlex. Data se stahovaly pomocí aplikace zvané Codeplex Downloader. Celá tato aplikace je napsaná v jazyce Python. Při její tvorbě byly použity standardní knihovny zaměřené zejména na oblast síťového programování a zpracovávání XML dokumentu.

V důsledku se jedná o více vláknový skript, který se snaží z cílového serveru pomocí paralelních dotazů stahovat textovou část obsahu dotazovaných stránek. Získaná data každé stránky mohou být uložena do separátních souborů. Vždy jsou předávána k dalšímu zpracování a to buď parseru HTML dokumentu nebo analyzátoru založeném na regulárních výrazech. Taktéž se používají ke zrychlené analýze založené na vyhledávání v částech stránky s konstantními offsety (původně však bylo napsáno pro akceleraci zpracování). To se však ukázalo být málo praktické ohledně striktní závislosti na konkrétním typu stránky. A pro dostatečně rychlého zpracování předešlými metodami se prakticky nepoužívá. Popis prohledávání stránek se uvádí v konfiguračním souboru, jehož sekce při interpretaci vytváří stromovou strukturu. V ní jsou uchovány podmínky přechodů a stahování dalších stránek prováděných nad nalezenými odkazy ve zpracovávané stránce. Struktura také nese popis a způsob ukládání dílčích elementů, které se ze stránky mají uchovat. Nalezené informace jsou uloženy v asociativním poli (slovník v jazyce Python) jehož obsah je ukládán podle definovaného předpisu do pomocných souborů (formát je proprietární a je optimalizován na rychlé procházení souboru podle indexu slovníku a stránky - identifikace generované při zpracovávání stránky na základě informací s konfiguračního souboru) a případně do databáze. Pomocné soubory slouží skriptu jen k orientaci ve struktuře stránek a pozdější kompletaci informací s ověřováním jejich korektnosti před dalším zpracováním. Zde běžně následuje uložení do databáze (finální dotaz ukládající záznam do databáze může být obvykle proveden až po zpracování více stahovaných stránek v kombinaci s dotazy na již existující data). Pro dotazování stránek ze serveru www.codeplex.com bylo použito knihovny urllib2, která umožňuje zejména snadnou tvorbu komplexních HTTP požadavků, ošetřuje výjimky při komunikaci se http serverem a interpretuje stavové kódy v odpovědích na HTTP dotazy.

Pro korektní zacházení s Cookies generovaných serverem byla použita Python knihovna pro paralelní zpracování a tedy nezávislé větve průchodu webem. XML parser stránek je postaven na knihovně libxml2dom a následné prohledávání ponejvíce na XPath. Regulární analýza stojí na knihovně re a vedle výrazově jednoduchého prohledávání stránek slouží ke zpracování non-html sekvencí.

Algoritmus stahování dat je tvořen následující smyčkou kroků:

1. generování URL stránky (zahrnuje GET i POST část požadavku)
2. stažení stránky a přiřazení elementu, který popisuje další zpracování

[12] Tato kapitola popisuje výsledky týmové práce, na které s námi spolupracovali další diplomanti a doktorand na katedře informatiky v období 2009 - 2010.

3. analýza XML parserem nebo regulárními výrazy nebo vhodnou kombinací obojího (na základě elementu s popisem zpracování)
4. vyhodnocení výsledků s jejich uložením do databáze nebo pomocných struktur

Postup získávání dat:

1. kompletace dat pro SQL INSERT z elementu s popisem zpracování
    a. prohledávání slovníků v paměti na nalezení hodnoty stanovené v elementu popisu
    b. prohledání pomocných souborů
2. zápis do databáze

Sumární informace o průběhu stažených dat je následující:
Doba stahování:

- celková kompletace řádově týdny
- optimalizovaný běh za méně než týden

Stahování prováděno iterativně:

- první krok dolování dat o projektech, uživatelích a vazbách mezi nimi. Součástí jsou dostupné
- informace o commit-ech a posledních prováděných akcích uživatelů
- druhý krok dolování diskuzí

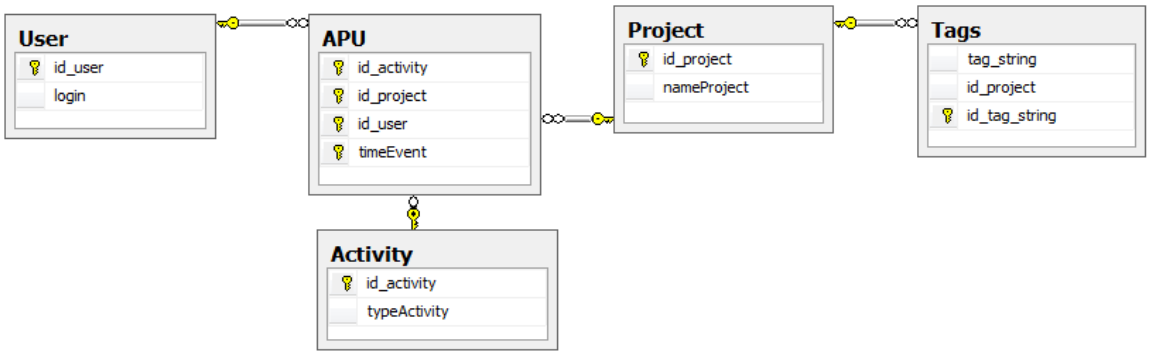
Zrychlení procesu dolování dat:

- masivnější paralelizmus - problém s potenciálně vysokou zátěží serveru CodePlexu
- optimálnější dotazování - důslednější znovupoužití získaných dat
- promyšlenější síťová komunikace - omezení množství dat přenášených sítí
- optimalizace na hranici problematiky útoků na server - využívání bezpečnostních mezer (syntetické inteligentní dotazování, optimalizace sekvencí dotazů, paralelní dotazování z více zdrojů)

Pro uložení všech stažených dat, byla navržena databáze *CodePlex*, která se skládá z tabulek:

- *User* (*login, personaStatemnet, createdOn, lastVisit, url, id_user*);
- *SourceCode* (*changeSet, login, nameProject, dowloand, date, time, id_commit*);
- *RecentActivity* (*login, nameProject, createdOn, type, activity, url, id_activity*);
- *Project* (*nameProject, tags, createdOn, status, license, pageViews, visits, url, id_project*);
- *Discussions* (*nameProject, dscTopic, login, insertTime, id_disc*).

Je zřejmě, že pro vytvoření samotné sítě uživatelů CodePlexu nebudeme potřebovat veškera data námi stažené. A výše popsaná struktura databáze není zrovna ideální pro další práci s data. Proto po zvážení byla vytvořena další vývojová databáze *codeplex_develop*, která byla určená přímo pro tvorbu sociální sítě a prací nad ní. Potřebná data byla importovaná pomocí SQL skriptů z předchozí databáze. Struktura nové finální vývojové databáze *codeplex_develop* je ukázána na obrázku 6.1.



Obrázek 6.1: Struktura vývojové databáze *codeplex_develop*

Nejdůležitější tabulkou pro nás zřejmě bude vazební tabulka *APU*, kde jsou evidované veškeré aktivity uživatelů. Aktivity mohou být různého typu, které se vedou v seznamu v tabulce *Activity*. Také se eviduje datum a čas kdy aktivita vznikla a v rámci kterého projektu byla provedena.

7. Implementace metody transformace bipartitního grafu sítě do unipartitních grafů

V této kapitole se pokusíme vysvětlit a popsat zvolenou metodu vytvoření a analýzy sociální sítě. Pro ukázku použijeme jen malou část dat s připravených pro vytvoření SN (viz. Kapitola 6). Experimenty nad celou sadou dat budou popsaní v další kapitole.

Fáze 1:

V první fázi vytvoříme bipartítní graf sítě. Také vysvětlíme, co jsou pro nás uzly a co pro nás znamenají hrany a váha hrany grafu. Pokusíme se popsat strukturu a postup vzniků bipartítního grafu, kterým začne analýza.

Před dalším popisem je nutno udělat menší úvod z teorie grafů o pojmu bipartitní graf. Pojmem bipartijní graf se v teorii grafů označuje takový graf, jehož množinu vrcholů je možné rozdělit na dvě disjunktní množiny tak, že žádné dva vrcholy ze stejné množiny nejsou spojeny hranou [5] .

Definice 7.1

Graf $G = (V, E)$ je bipartitní, pokud platí
$V = V_1 \cup V_2, V_1 \cap V_2 = \emptyset$ a $\forall e = \{u, v\}, e \in E: u \in V_1 \, a \, v \in V_2$

Platí-li navíc $E = V_1 \times V_2$ (tedy v grafu existují všechny hrany s touto vlastností), nazývá se tento graf úplný bipartitní. Značí se $K_{m,n}$, kde m a n jsou velikosti obou partit [5] .

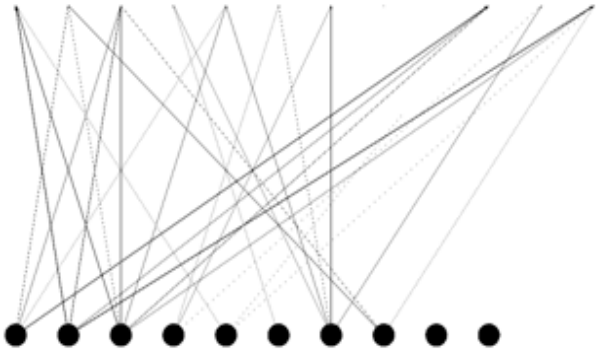
Vlastnosti [5] :

- obě partity grafu jsou podle definice nezávislé množiny a graf přímo implikuje jedno možné 2-obarvení
- platí i obrácené tvrzení – všechny 2-barevné grafy jsou bipartijní
- jednoduchým algoritmem lze v lineárním čase zjistit, zda je daný graf bipartitní a také nalézt jeho 2-obarvení (průchodem do hloubky)
- každý strom je bipartijní
- graf je bipartitní právě tehdy, neobsahuje-li kružnici liché délky

K-partitní graf

Pojem bipartitnosti lze zobecnit na libovolné $k \geq 2$. Je-li $G = (V, E)$ graf a V lze rozložit na k disjunktních podmnožin takových, že žádné dva vrcholy ze stejné podmnožiny nejsou spojeny hranou, pak tento graf nazýváme k-partitním grafem. Je-li tento graf úplný (ve stejném smyslu jako úplný bipartitní graf, viz výše) a počty vrcholu v jednotlivých partitách jsou $n1, n2, \ldots\ldots, nk$, pak se tento graf značí $K_{n1, n2, \ldots\ldots, nk}$ a nazývá se úplný k-partitní graf [5] .

Příklady vzniklého bipartintního grafu jsou zobrazené na obrázků níže. Graf se skládá z dvou skupin uzlů. Skupina uzlů nahoře v grafu reprezentuje uživatele, dole - projekty. Hrana anebo vazba mezi uživatelem a projektem znamená, že uživatel zúčastnil projektu, s kterým je spojen, jinak řečeno – uživatel provedl, aspoň jednu aktivitu na tomto projektů. Váha anebo tloušťka hrany říká nám o tom, kolik aktivit uživatel provedl v rámci projektu.



Obrázek 7.1: Bipartitní graf cele sítě projektů a uživatelů

Fáze 1a – rozšíření:

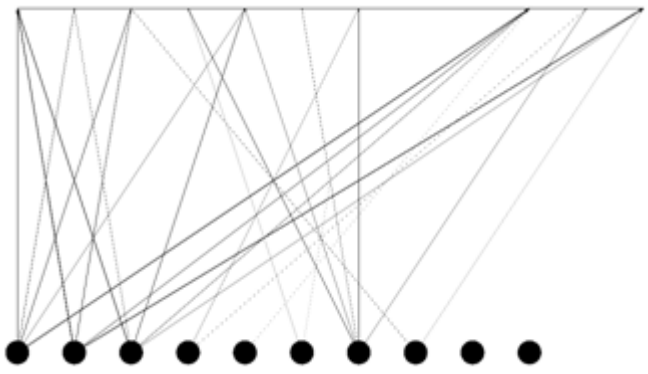
Logiku vytvoření bipartitního grafu uživatelů a projektů, můžeme rozšířit o dynamickou složku. Což neznamená nic jiného než vytvoření grafu, který odpovídá realitě v určitém časovém úseku. Časový úsek může být libovolné délky. Rozdíl mezi standardním postupem vytvoření bipartitního grafu popsanému ve fázi 1 a tímto rozšířením je pouze filtrování aktivit uživatelů dle data vzniku této aktivity. Jako výsledek vzniká bipartitní graf uživatelů a projektů pro určité zadané časové období.

Fáze 1b – rozšíření:

Dalším možným rozšířením pro vytvoření bipartitního grafu uživatelů a projektů je rozlišení vazeb dle typu aktivity. Což znamená vytvoření grafu s vazbami pouze určitého typu aktivity. Toto rozšíření je možno kombinovat s rozšířením popsaném ve fázi 1a.

Fáze 2:

Ke grafu s obrázku 7.1 (Bipartitní graf celé sítě projektů a uživatelů) přidáme vazby mezi jednotlivými uživateli. Hrana mezi uživateli vznikne pouze tehdy, pokud tito dva uživatelé mají alespoň jeden společný projekt, jinak řečeno - prováděli aktivity v rámci společného projektu. Potom váha (tloušťka) nově vytvořené hrany mezi uživateli se určí počtem společných projektů mezi nimi. Výsledný graf je zobrazen na obrázku 7.2.



Obrázek 7.2: Graf celé sítě projektů a uživatelů doplněn o vazby mezi uživateli

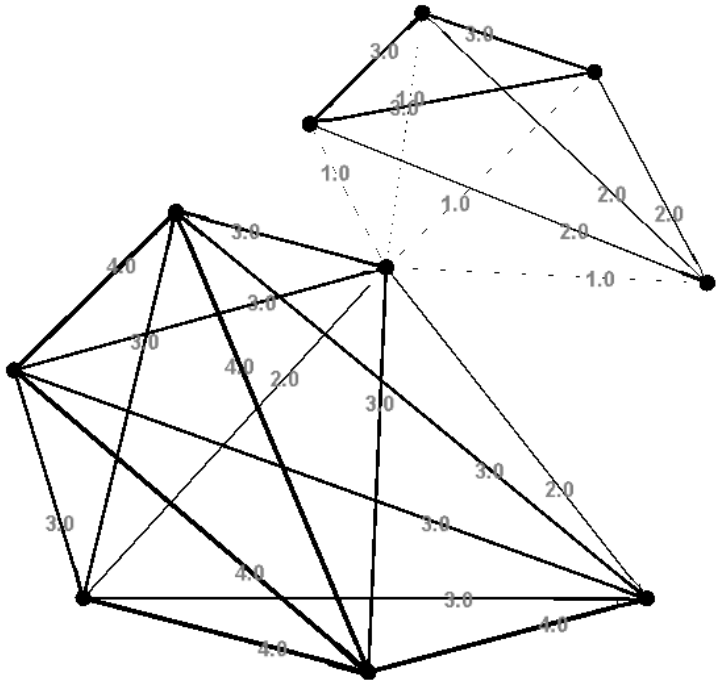
Fáze 3:

V této fázi proběhne samotná transformace bipartitního grafu do unipartitního grafu, který obsahuje pouze uzly typu uživatel, a to pomocí odstraněni uzlů typu projekt a hran k nim patřícím.

Jako výsledek této fáze může vzniknout buď síť skládající se z mnoha komponent mezi sebou nepropojených (případ většinou menší sítě, kde jsou jasně viditelní komunity) anebo jedna souvislá komponenta.

Pod každou komponentou se nyní rozumí neorientovaný ohodnoceny graf uživatelů.

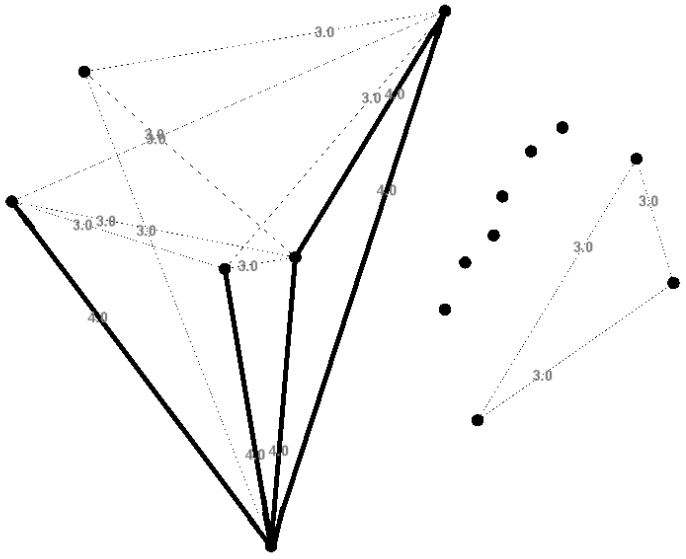
V případě, že jako výsledek transformace vznikne jedna velká souvislá komponenta, jak je to zobrazeno na obrázku 7.3, je zapotřebí pro další práci s grafem a komunitami v něm navrhnout metodu rozřezaní výsledného grafu sítě.



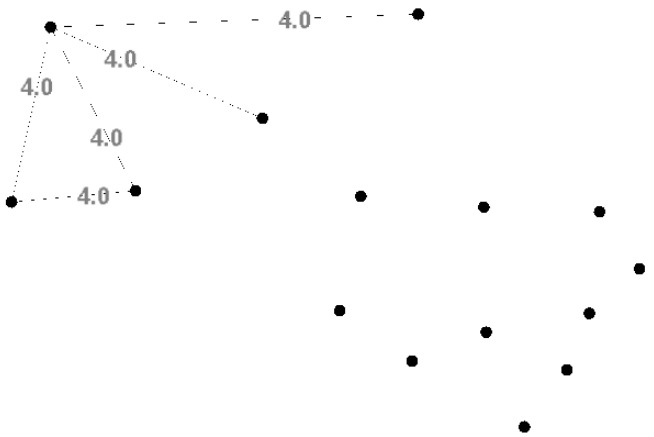
Obrázek 7.3: Příklad vzniku jedné souvisle komponenty. Váha = 1 (počet společných projektů)

Fáze 4:

V této fázi popisujeme metody rozdělení sítě na menší komponenty. Jednou z možných metod rozděleni grafu sítě je zanedbání některých hran grafu sítě na základě jejich váhy, která v našem případě určují počet společných projektů. V důsledku implementace tohoto filtrování "zmizí" z grafu sítě nepodstatné anebo méně podstatné hrany a se zmenšením propojenosti se síť může rozpadnout na několik souvislých komponent. Příklad takového rozkladu je zobrazen na obrázku 7.4. Na druhu stranu není korektní používat tuto metodu na menších grafech, kde každá z hran může být podstatná (s ohledem na menší počet projektů v této menší síti). Ale ani na velkých grafech nemusí vždy uvedená metoda vést k očekávanému výsledku a graf může zůstat jedna souvislou komponentou. V budoucnu lze zde uvažovat o použití spektrálního rozkladu k dělení této sítě [2] .



Obrázek 7.4: Váha = 3 (přiklad úspěšného rozděleni na komunity)



Obrázek 7.5: Váha = 4 (přiklad úspěšného rozděleni na komunity)

Dále uvádíme pseudokód kompletního algoritmu vytvoření sociální sítě, kde jako vstupní parametry jsou objekt CodePlex a minimální počet společných projektů uživatelů pro vytvoření hrany mezi nimi (viz. Algoritmus 7.1). Na výstupu algoritmu je očekáván výsledný graf sociální sítě.

```
Transformace(codeplex, int index)
Graph g;
Forech(user in users)
{
      g.addNode(user.Id, user.Login);
}
Forech(user in users)
{
      Dict commons<user, hash<project>>  user € users
      Dict projectsOfuser <project, int> = user.getProjects()
      Forech(project in projectsOfUser.Keys)
```

```
        {
              Forech(otherUserOfProject in project.GetUsers())
              {
                    If(otherUserOfProject != user)
                    {
                          If(commons.Keys.Contains(otherUserOfProject)
                                commons[otherUserOfProject].Add(project);
                    }
              }

        }
        Forech(commonUser in commons.Keys)
        {
              If(commons[commonUser].Count>0 and
              commons[commonUser].Count>=index)
              {
                    g.addEdge(user, commonUser, commons[commonUser].Count,
                    commons[commonUser].projectsNames);
              }
        }
}
Return g;
```

Algoritmus 7.1 (Transformace bipartitního grafu sítě do unipartitních grafů)

8. Experimenty

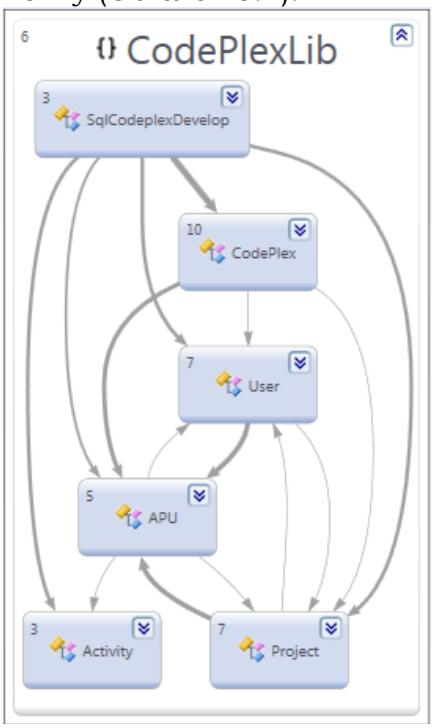
V kapitole 6 je popsáno, jakým způsobem získáváme data z CodePlexu a kam data ukládáme. Před popisem provedených experimentů si ještě něco více řekneme o obsahu vstupních dat. Tabulky databáze *codeplex_develop*, nad kterými budeme pracovat, jsou naplněny následujícím způsobem:

Tabulka	**Počet záznamů/řádku**
User	31614
Project	27640
Activity	1
APU	238580
Tags	84476

Tabulka 8.1: Naplnění tabulek databáze *codeplex_develop*

Pro vytvoření sociální sítě budeme potřebovat tabulky *User*, *Project*, *APU* a *Activity*. Tabulka *Tags*, která v budoucnu bude sloužit, jako pomocná tabulka pro vyhledávaní projektů dle tágu v rámci cele sítě, pro tvorbu grafu sociální sítě není zajímavá. Je důležité upozornit, že prozatím máme zavedený pouze jeden druh aktivity - "*SourceCode*". Všechny registrované aktivity z tabulky *APU* jsou tohoto typu.

Pro práci s daty a jejich transformace do bipartitního a následně unipartitního grafu, byl zvolen objektově orientovaný přístup. Pro prezentaci dat a práci s nimi jako objekty, byla vytvořena knihovna *CodePlexLib*. Níže uvádíme diagram výše zmíněné knihovny (Obrázek 8.1).

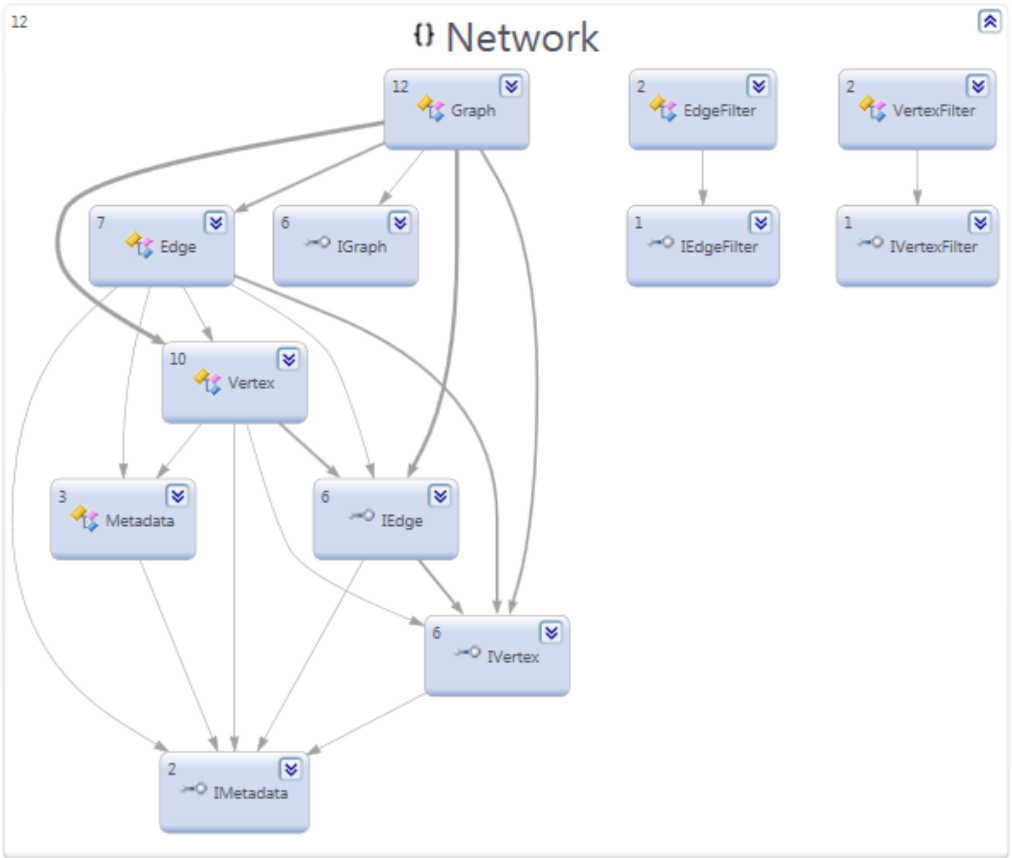


Obrázek 8.1: Diagram knihovny *CodePlexLib*

Z třídního diagramu vidíme, že pro reprezentaci každého z objektů byla vytvořena třída, pomocí které definujeme jeho vlastnosti. Třída *SqlCodeplexDevelop* slouží pro práci s databází a samotné načtení dat do příslušných objektů.

Třída *CreateGraphUsers,* byla umístěna mimo tuto knihovnu pro možnost budoucího využití pro jiné typy vstupních dat. Tato třída slouží pro vytvoření sociální sítě standardním způsobem nebo s použitím rozšíření popsaných v kapitole 7 fáze 1a a 1b, pro možnost budoucího rozšíření a možného využití pro jiné typy vstupních dat.

Pro práci s výsledným grafem, teda unipartitním grafem uživatelů, byla naimplementována knihovna *Network*. *Network* reprezentuje prvky sítě jako grafové prvky. Myslíme tím vrcholy, hrany a jejich vlastnosti (id, váha hrany, jména uzlů atd.). Knihovna *Network* byla navrhnutá hlavně jako prostředí pro práci s neorientovaným ohodnoceným grafem. Graf, který nám vznikne jako výsledek transformace do unipartitního grafu, splňuje tyto vlastnosti. Ale i přes svůj hlavní účel se dá knihovna *Network* použít i jako nástroj pro práci s neorientovaným neohodnoceným grafem a po menších úpravách i pro práci s orientovanými grafy. Pro snadnější a rychlejší práci s objekty grafu sítě, byly naimplementovány některé metody pro manipulaci s prvky a jejich vlastnostmi (například: metoda pro získání všech sousedů vrcholu, všech přilehlých hran, celkový počet všech uzlů nebo hran sítě, atd.) Dále uvádíme diagram knihovny *Network* a stručný popis jednotlivých tříd a jejich vlastností.



Obrázek 8.2: Diagram knihovny Network

Metadata – třída pro reprezentaci doplňujících informací k objektu.

Vetrex – třída reprezentující uzel v síti. Vlastnosti třídy jsou: jednoznačné id uzlu, seznam přilehlých hran, *Metadata*. Metody třídy jsou: přidání hrany a vyhledávání sousedních uzlů.

Edge – třída reprezentující hranu v síti. Vlastnosti třídy jsou: jednoznačné id hrany, VertexA a VertexB – uzly, které tato hrana spojuje, váha hrany, *Metadata*.

Graph – třída reprezentující graf sítě. Vlastnosti třídy jsou: seznam všech uzlů a hran, celkový počet hran. Metody třídy jsou: Přidání uzlu, hrany, vyhledávání uzlu podle id, výpočet celkového počtu uzlů nebo hran.

EdgeFilter a *VertexFilter* – třídy pro budoucí rozšíření průchodů a vyhledávání v grafu sítě.

Pro uložení, znovu načtení mezí výsledků a případnou transformaci grafu do jiných formátů pro další práci s nimi, byly naimplentovány metody pro export a import grafu. Z hlediska toho, že pro vizualizaci grafu, resp. podgrafu byl použít nástroj Gephi, byly připraveny metody transformace do a z následujících formátů: graphml, csv_nodes a csv_edges pro gephi, jednoduché xml a csv formáty pro reprezentaci grafu.

Dále byla naimplementována knihovna *CentralityLib*. Tato knihovna obsahuje třídy pro výpočet základních centralit, které jsme popisovali v podkapitole 4.3. Jako vstup pro výpočet každé z centralit je brán objekt *Graph* a *Vertex*, pro který potřebujeme vybranou centralitu vypočítat. Také byly připravené metody transformace mezi objektem *Graph* a maticí sousednosti. Pro tyto účely byla vytvořena třída *MatrixVsGraph*.

Definice8.1: Matice sousednosti (nebo také incidenční matice) je čtvercová matice, která má na pozici *ij* nulu pokud mezi *i*-týma *j*-tým uzlem neexistuje hrana. Pokud mezi *i*-tým a *j*-tým uzlem hrana existuje je *ij*-tá pozice obsazena hodnotou váhy hrany [5] .

Příklad matice sousednosti grafu o 4 vrcholech:

	u1	u2	u3	u4
u1	0	2	0	0
u2	0	0	0	1
u3	1	1	0	1
u4	0	0	0	0

Hledaní nejkratší cesty ke každému z vrcholů, hledaní všech nejkratších cest v grafu a podobné pomocné algoritmy pro výpočet centralit pracují právě s grafem reprezentovaným pomocí matice sousedností. A to hlavně pro snížení časové složitostí výpočtů algoritmů. Také pro tento účel se používá řídká matice reprezentující graf.

K nalezení nejkratší cesty v grafu byl použit známý Dijkstrův algoritmus.

Dijkstrův algoritmus je algoritmus sloužící k nalezení nejkratší cesty v grafu. Je konečný (pro jakýkoliv konečný vstup algoritmus skončí), protože v každém průchodu cyklu se do množiny navštívených uzlů přidá právě jeden uzel. Průchodů cyklem je tedy nejvýše tolik, kolik má graf vrcholů. Funguje nad hranově kladně ohodnoceným grafem [5] (pseudokód je uveden ve výpisu algoritmu 8.1).

```
Function Dijkstra(G, w, s)
      for each vertex v in V[G]   // Initialization
         do d[v] := infinity
            previous[v] := undefined
      d[s] := 0
      S := empty set
      Q := set of all vertices
      while Q is not an empty set
         do u := Extract-Min(Q)
           S := S union {u}
           for each edge (u,v) outgoing from u
              do if d[v] > d[u] + w(u,v) //Relax( u,v)
                 then d[v] := d[u] + w(u,v)
                    previous[v] := u
```

Algoritmus 8.1 (Dijkstrův algoritm)

8.1 Experiment 1

Krok 1

V prvním kroku jsme naplnily daty z databáze *codeplex_develop* instance tříd *CodePlex*. Čímž nám vznikla objektová reprezentace projektů, uživatelů, aktivit a vazeb mezi nimi. V tuto chvíli jsme dostali k dispozici nepřímou reprezentaci bipartitního grafu sítě (pomocí vazeb mezi projekty a uživateli. Komunikaci mezi uživateli, kterou používáme, myslíme komunikaci z pohledu SN, ale reálně je to spolupráce na projektech.

Krok 2

V dalším kroku cestou transformace bipartitního grafu do unipartitního grafu, jsme vytvořili instanci třídy *Graph* s příslušnými hranami, uzly a jejich vlastnostmi. Minimální akceptovaná váha hrany (počet společných projektů) je 1. Vzniklý graf je ohodnocený a neorientovaný, kdy ohodnocení hran bylo stanoveno počtem společných projektů mezi dvěma uživateli. Seznam názvů společných projektů a login uživatele jsou uloženy jako *Metadata* k příslušným objektům.

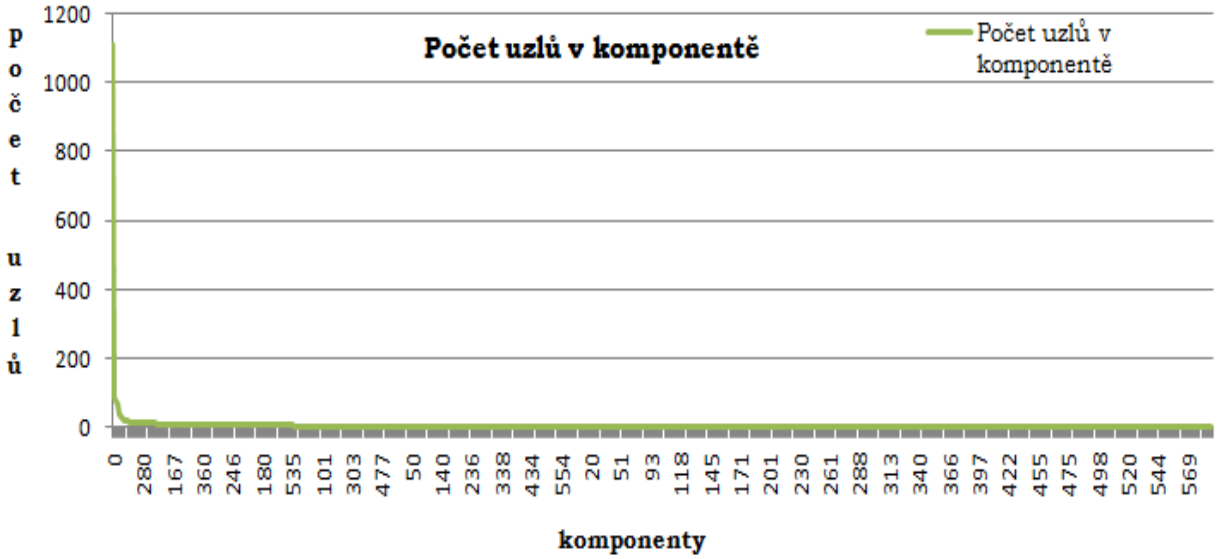
Výsledný graf obsahuje 31614 vrcholů, což by mělo a odpovídá celkovému počtu uživatelů CodePlexu. Celkový počet hran ve výsledném grafu je 47324. Tento graf byl uložen ve všech formátech (graphml, csv pro gephi, jednoduchá reprezentace grafu v csv a xml).

Výsledný graf sítě byl zobrazen v nástroji Gephi, který byl popsán v podkapitole 4.2. Data do tohoto nástroje byly naimportované ve formátu graphml. Následně byly použity některé z vnitřních metod Gephi pro vylepšení přehlednosti zobrazení velkých grafů. Obrázek se zobrazením grafu najdete v příloze A.

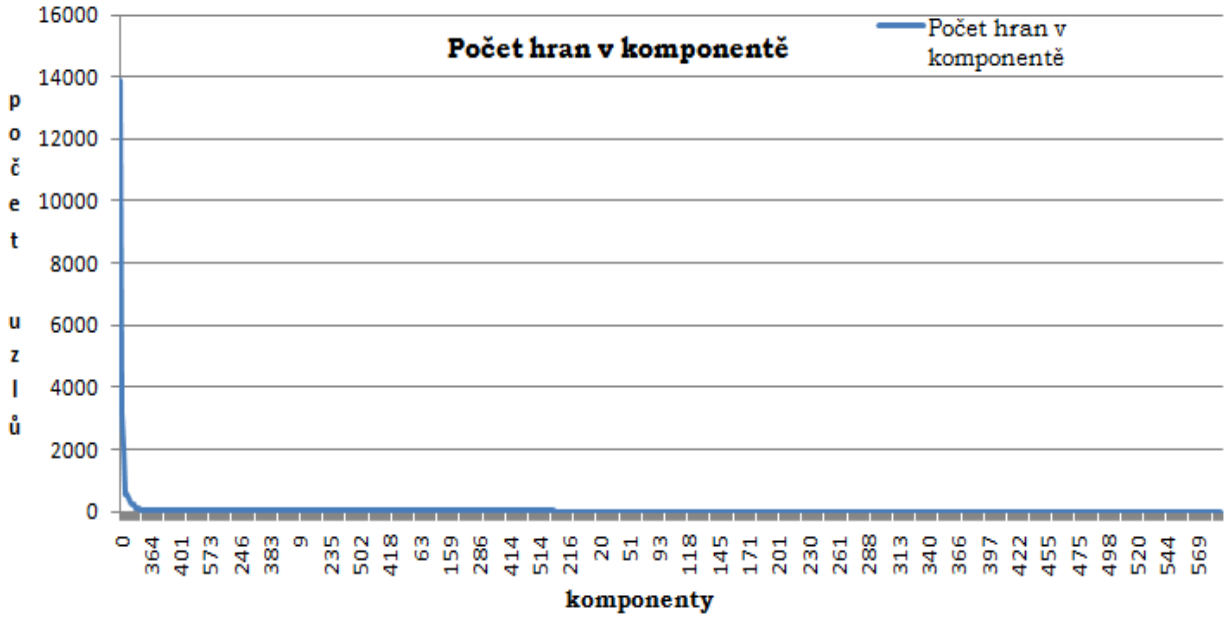
Krok 3

V grafu, který vznikl v předchozím kroku, byly vyhledány všechny souvislé komponenty, které jsme následně uložili, jako samostatné grafy do objektů typu *Graph*. Transformace nalezených komponent do objektu *Graph*, nám přináší řadu výhod pro další práci s každou z nich: export do různých formátů, následná vizualizace, možnost výpočtu hodnot centralit a aj. V síti CodePlex bylo nalezeno

celkem 590 souvislých komponent různé velikosti. Na znázornění 8.1.1 a 8.1.2 jsou zobrazené diagramy počtu vrcholů a hran v každé z komponent.



Obrázek 8.1.1: Diagram počtů uzlů v komponentách



Obrázek 8.1.2: Diagram počtů hran v komponentách

Čím víc uzlů má komponenta, tím větším počtem hran je propojena, což vidíme z výše uvedených diagramů. Velikosti jednotlivých komponent rozebereme něco důkladněji, a to s pomocí tabulky 8.1.1, kde uvádíme celkový počet komponent o různých velikostech.

Počet uzlů	Počet komponent
2	345
3	95
4	53
5	24
6	17
7	13
8	4
9	9
10	6
11	5
12	5
13	1
14	3
15	1
17	1
21	1
22	1
34	1
37	1
65	1
86	1
1112	1

Počet hran	Počet komponent
1	345
2	12
3	86
4	14
5	2
6	39
7	4
9	1
10	16
11	6
12	1
13	2
14	1
15	9
16, 18	po 2
19	1
20	2
21	11
22	2
24, 26, 27	po 1
28, 29	po 2
36	5
37, 38, 39	po 1
45, 55	po 2
57, 61	po 1
66	2
91, 193, 210	1
276, 561, 2080	1
3109	4
13861	1

Tabulka 8.1.1: Tabulka počtů komponent dle počtů uzlu v nich.

Největší z nalezených komponent, je komponenta o 1 112 vrcholech, propojených mezi sebou 13 861 hranami. Další komponenty jsou mnohem menší, dokonce většina z nich jsou komponenty o dvou vrcholech. Celkový počet komponent složených pouze z dvou vrcholů je 345, z třech – 95 a z čtyř – 54. 67 dalších komponent se pohybují v rozmezí od 5 do 10 vrcholů. Velikosti zbylých 28 komponent jsou v rozmezí od 10 do 86 vrcholů. Průměrný počet uzlů v komponentech je 5,552542373 a průměrný počet hran je 40,1050847.

Po důkladnějším rozboru komponent, byly odhalené často vyskytující se typy struktury komponent. Pro rozpoznání těchto základních typů struktur, byla naimplementovaná třída *StructurAnalyze*. Pro menší komponenty o dvou nebo třech vrcholech, bylo navrhnuté vlastní rozdělení.

Rozdělení komponent dle struktury:
Dvojice slabá - komponenta tvořena dvěma vrcholy, které jsou spojené hranou s váhou 1.
Dvojice silná - komponenta tvořena dvěma vrcholy, které jsou spojené hranou s váhou větší než 1.
Trojice – přímka - komponenta tvořena třemi vrcholy, kde pouze jeden z nich je spojen s oběma dvěma ostatníma.
Trojice - hvězda – komponenta tvořena třemi vrcholy, kde každý z nich je spojen se všemi ostatními vrcholy.
Hvězda klasická - komponenta tvořena více než třemi vrcholy, kde každý uzel je spojen s každým dalším uzlem hranou, kdy váha všech hran je 1.
Hvězda s podskupinou - komponenta tvořená více než třemi vrcholy, kde každý uzel je propojen s každým dalším uzlem hranou. Váhy hran se rovnají nebo jsou vetší než 1, kdy alespoň jedna hrana je ohodnocena hodnotou větší než 1.
Hvězda plus jedinec - hvězda libovolného typu (klasická nebo s podskupinou) kdy jeden uzel je propojen pouze s jedním uzlem z hvězdy.
Dvě hvězdy – dvě hvězdy libovolného typu (klasické nebo s podskupinou) a jeden uzel, kdy jedině on je propojen se všemi uzly z obou hvězd.

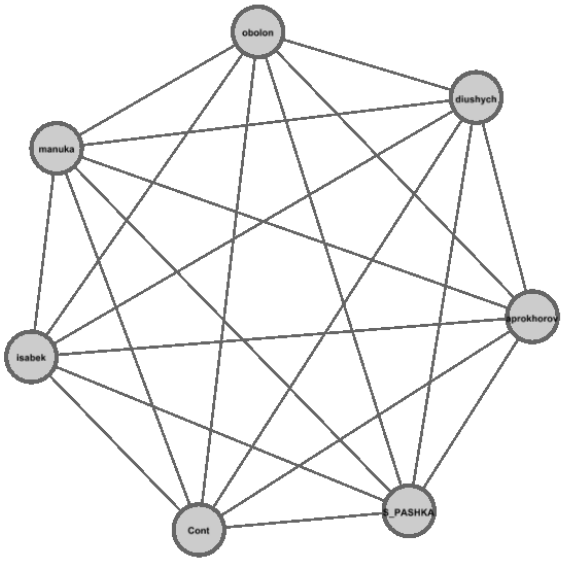
Typ struktury	**Počet komponent**
Dvojice slabá	340
Dvojice silná	5
Trojice – přímka	12
Trojice – hvězda	83
Hvězda klasická	83
Hvězda s podskupinou	3
Hvězda plus jedinec.	27
Dvě hvězdy	7
Jiný	30

Tabulka 8.1.2: Rozdělení komponent dle struktury

Krok 4

V dalším kroku, bylo zvoleno několik komponent různé struktury pro ukázku důkladnější analýzy těchto komponent jako komunit. Pomocí již zmíněné knihovny *CentralityLib*, pro vyhodnocení významu každého z vrcholů v komponentech, byly vypočteny jejich centrality. Grafy komponent, byly vizualizované pomocí nástroje Gephi.

Komponenta číslo 7 - Hvězda klasická (viz. Obrázek 8.1.3)
Počet vrcholů: 7
Počet hran: 21
Počet hran s hodnotou váhy 1: 21



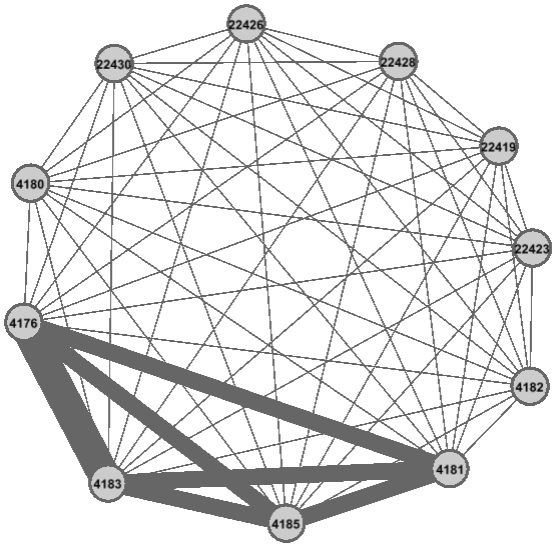
Obrázek 8.1.3: Komponenta č. 7

Komponenta číslo 7 reprezentuje typický příklad skupiny lidí, vzájemně spolupracujících pouze na jednom jediném projektu, v tomto konkrétním případě: „Protoforma | Tactica Adversa“. Na první pohled můžeme vyčíst z obrázku, že každý z uživatelů na tomto projektu komunikoval s každým jiným účastníkem, se stejnou intenzitou a všichni účastníci mají shodné vlastnosti. Tuto myšlenku potvrzují výsledky výpočtů centralit pro každý z uzlů, které uvádíme v tabulce 8.1.3. Hodnoty centralit u každého z uzlů jsou shodné a to znamená, že žádný z nich se nevyčleňuje svými vlastnostmi.

Degree	**Closeness**	**Betweenness**	**Eigenvector**
1	0,166666667	0	0,3779644730

Tabulka 8.1.3: Hodnoty centralit pro všechny uzly komponenty č. 7

Komponenta číslo 223 - Hvězda s podskupinou (viz. Obrázek 8.1.4)
Počet vrcholů: 11
Počet hran: 55
Počet hran s hodnotou váhy 1: 19
Počet hran s hodnotou váhy 2: 5
Počet hran s hodnotou váhy 3: 1



Obrázek 8.1.4: Komponenta č. 223

Struktura komponenty č. 223 je nápodobná předchozí komponentě. Všichni účastníci spolupracovali na projektu "The Open Source Phasor Data Concentrator", ale 4 účastníci se spojili navíc v projektu "Time Series Framework" a dva z nich pracovali společně na třetím projektu "TVA Code Librery". Vazba mezi těmito účastníky a jejich role v této komponentě, bude významnější, než u jiných účastníků, což potvrzují vysoké hodnoty eigenvector centralit těchto uzlů. Výsledné hodnoty centralit jsou uvedené v tabulce 8.1.4.

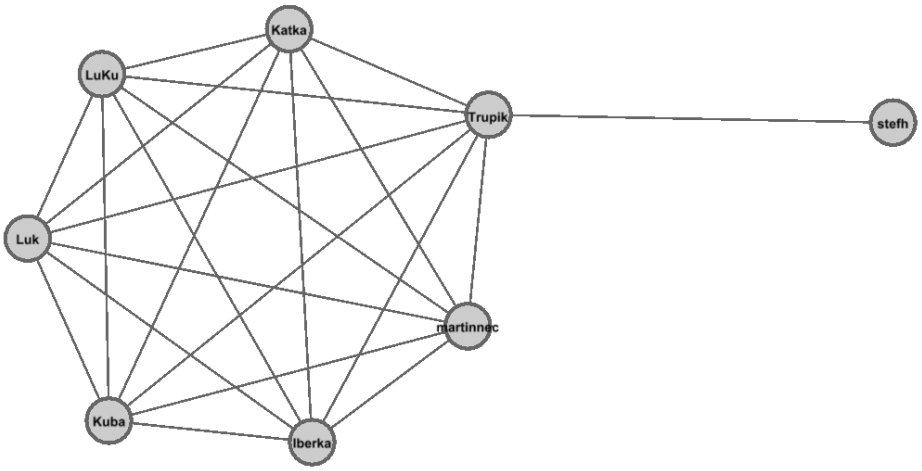
Id vrcholu	Login	Degree	Closeness	Betweenness	Eigenvector
4176	ritchiecarroll	1	0,076923077	0	0,375832726
4180	mbrahmbhatt	1	0,1	0,75	0,260531515
4181	mthakkar	1	0,076923077	0	0,348111212
4182	paul_trachian	1	0,1	0,75	0,260531514
4183	pinalpatel	1	0,076923077	0	0,375832726
4185	staphen	1	0,076923077	0	0,348111212
22419	andyhill	1	0,1	0,75	0,260531515
22423	galenriley	1	0,1	0,75	0,260531515
22426	jpatterson	1	0,1	0,75	0,260531515
22428	nischal	1	0,1	0,75	0,260531515
22430	ryanzuo	1	0,1	0,75	0,260531515

Tabulka 8.1.4: Hodnoty centralit pro každý z uzlů komponenty č. 223

Průměrná hodnota degree centrality v komponentě 223 je 1, closeness centrality: 0,091608, betweenness centrality: 0,477273, eigenvector centrality: 0,297419.

Komponenta číslo 267- Hvězda plus jedinec (viz. Obrázek 8.1.5)
Počet vrcholů: 8
Počet hran: 22

Počet hran s hodnotou váhy 1: 22



Obrázek 8.1.5: Komponenta č. 267

Komponenta č. 267 představuje další zajímavý typ struktury vyskytující se u několika nalezených komponent. Část komponenty je představena již dříve zmíněnou hvězdou. V našem případě všichni uživatelé ve hvězdě navzájem spolupracovali na projektu " XCase - Tool for XML Data Modeling". Výjimka je v tom, že jeden z účastníků (Trupik, id - 5061) pracoval navíc s dalším uživatelem (stefh, id-5049) na projektu " CommandLine Parser Library", což říká o jeho výši aktivitě v porovnání s ostatními uzly. Naopak účastník stefh se tváří v této komponentě jako osamocený a málo komunikativní, proto že už nemá žádná další spojení. Tyto vlastnosti potvrzují výsledky výpočtu centralit uvedené v tabulce 8.1.5.

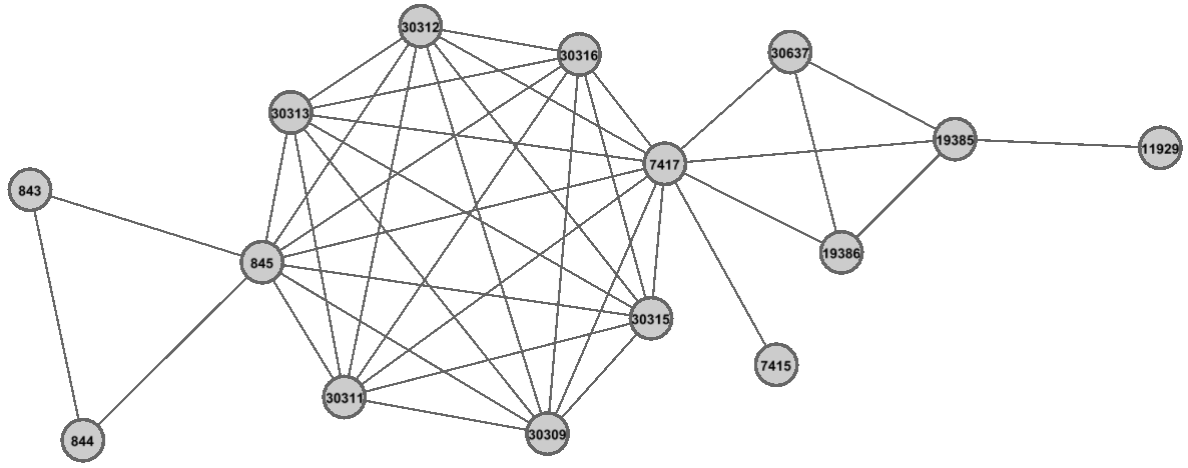
Id vrcholu	Login	Degree	Closeness	Betweenness	Eigenvector
5049	stefh	0,142857143	0,076923077	0	0,063905
5061	Trupik	1	0,142857143	6	0,384977
5059	Kuba	0,857142857	0,125	0	0,375879
5060	Luk	0,857142857	0,125	0	0,375879
5063	lberka	0,857142857	0,125	0	0,375879
5064	martinnec	0,857142857	0,125	0	0,375879
31562	LuKu	0,857142857	0,125	0	0,375879
31563	Katka	0,857142857	0,125	0	0,375879

Tabulka 8.1.5: Hodnoty centralit pro každý z uzlů komponenty č. 267

Průměrná hodnota degree centrality v komponentě 267 je 0,785714, closeness centrality: 0,121223, betweenness centrality: 0,75, eigenvector centrality: 0,338019.

Komponenta číslo 51 – Jiný typ struktury (viz. Obrázek 8.1.6)
Počet vrcholů: 15
Počet hran: 39

Počet hran s hodnotou váhy 1: 39



Obrázek 8.1.6: Komponenta č. 51

Komponenta číslo 51 se skládá z třech hvězd o velikosti 3, 4 a 8 uzlů. Někteří účastníci spolupracovali i na dalších projektech s jinými účastníky. Celkem se dá říct, že vlastnosti jednotlivých uzlů této komponenty nejsou úplně odhalitelné na první pohled. Proto se analýza této komponenty, provede přímo s pomocí centralit uzlů ukázaných v tabulce 8.1.6.

Id vrcholu	**Login**	**Degree**	**Closeness**	**Betweenness**	**Eigenvector**
843	joshtwist	0,142857143	0,03125	0	0,05814529
844	Morgan	0,142857143	0,03125	0	0,05814529
845	sleeks	0,642857143	0,05	24	0,357015992
7417	Melborp	0,785714286	0,058823529	49	0,376448833
7415	kaidokert	0,071428571	0,033333333	0	0,052723429
19385	gpeipman	0,285714286	0,038461538	13	0,074774959
11929	phertzog	0,071428571	0,025641026	0	0,010472585
19386	hampsu	0,214285714	0,037037037	0	0,073488412
30637	jevgeni	0,214285714	0,037037037	0	0,073488412
30309	edglas	0,5	0,045454545	0	0,342729798
30311	lkruger	0,5	0,045454545	0	0,342729798
30312	mahipalk	0,5	0,045454545	0	0,342729798
30313	mtaute	0,5	0,045454545	0	0,342729798
30315	yutong	0,5	0,045454545	0	0,342729798
30316	gbhanda	0,5	0,045454545	0	0,342729798

Tabulka 8.1.6: Hodnoty centralit pro každý z uzlů komponenty č. 51

Největší hodnotu degree centrality mají uživatelé 7417 a 845, z čehož jsme schopni posoudit, že s velkou pravděpodobností se jeví tito účastníci jako nejpopulárnější v této komponentě. A naopak nejméně komunikativním je účastník 11929 s nejmenší hodnotou degrese centrality.

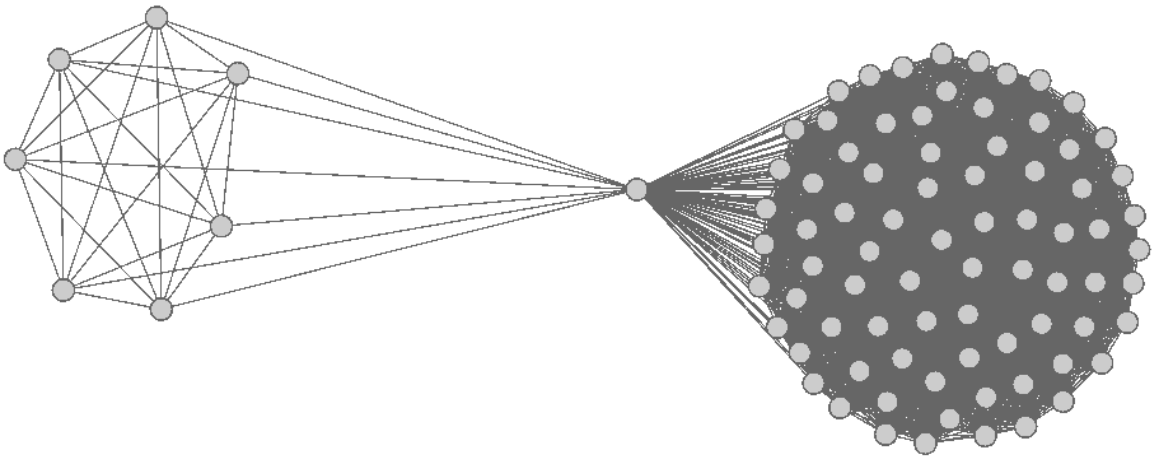
Hodnoty closeness centralit jsou skoro stejné u všech uzlů, což znamená, že uzly mezi sebou v rámci této komponenty komunikují skoro se stejnou rychlostí a jsou v poměrně stejné vzdálenosti. Nejlepší podmínky pro dostupnost má uzel 7415, který má nejvyšší hodnotu closeness centrality.

Vysoké hodnoty betweenness centrality mají uzly 845, 7417 a 19385. To znamená, že tito uživatelé jsou velmi důležití pro propojení jiných účastníků této komponenty.

Největší hodnoty eigenvector centrality mají uživatelé 7417 a 845 a v souhlasu s významem této centrality můžeme říct, že tito uživatelé jsou nejvýznamnější v komponentě 51. Což vlastně odpovídá i výsledkům odvozených z centralit betweenness a degree.

Průměrná hodnota degree centrality v komponentě 51 je 0,371428571, closeness centrality: 0,041037385, betweenness centrality: 5,733333, eigenvector centrality: 0,212739.

Komponenta číslo 464 - Dvě hvězdy (viz. Obrázek 8.1.7)
Počet vrcholů: 86
Počet hran: 3109
Počet hran s hodnotou váhy 1: 3109



Obrázek 8.1.7: Komponenta č. 464

Komponenta 464 se skládá z dvou skupin lidí pracujících na dvou různých projektech. Vlevo na obrázku můžeme vidět menší hvězdu tvořenou 7 účastníky nad projektem „EnhSim“, vpravo vetší hvězdu o 78 účastnících pracujících spolu na projektu „Rawr“. Nejzajímavější účastník této komponenty je účastník s id 17114. Účastník 17114 je umístěn uprostřed mezi těmito dvěma hvězdami což znamená, že se zúčastnil obou dvou projektů a může sloužit jako propojovací prvek mezi těmito dvěma skupinami účastníků. Jeví se pravděpodobně jako nejkomunikativnější a nejpopulárnější uzel v celé komponentě. Navíc má zřejmě nejlepší podmínky pro rychlou komunikaci s ostatními účastníky. Naše tvrzení můžeme odůvodnit, pomocí výpočtu centralit pro všechny uzly celé komponenty. Výsledky uvádíme v tabulce 8.1.7, kde 7 vrcholů reprezentují hodnoty centralit všech vrcholů levé skupiny, 78 vrcholů – hodnoty centralit všech vrcholů pravé skupiny a hodnoty centralit propojující účastníka 17114 se uvádí zvlášť.

Vrcholy	Degree	Closeness	Betweenness	Eigenvector
id=17114	1	0,011764706	546	0,112644693
78 vrcholů	0,917647059	0,010869565	0	0,11250607
7 vrcholů	0,082352941	0,006134969	0	0,001564483

Tabulka 8.1.7: Hodnoty centralit uzlů komponenty č. 464

Průměrná hodnota degree centrality v komponentě 464 je 0,850615595, closeness centrality: 0,0104946, betweenness centrality: 6,348837209, eigenvector centrality: 0,103477553.

8.2 Experiment 2

Krok 1 a 2

Jako druhý menší experiment byla zvolena metoda redukce grafu, zmíněná na konci kapitoly 7. Tato metoda spočívá ve vytvoření grafu sítě cestou ořezání nepodstatných vazeb. Na vstupu byly stejná data jako v předchozím experimentu, ale minimální hodnota vazby mezi uzly byla stanovena na hodnotu 2. To znamená, že pokud existuje vazba mezi dvěma uživateli v grafu sítě, podíleli se spolu alespoň na dvou projektech.

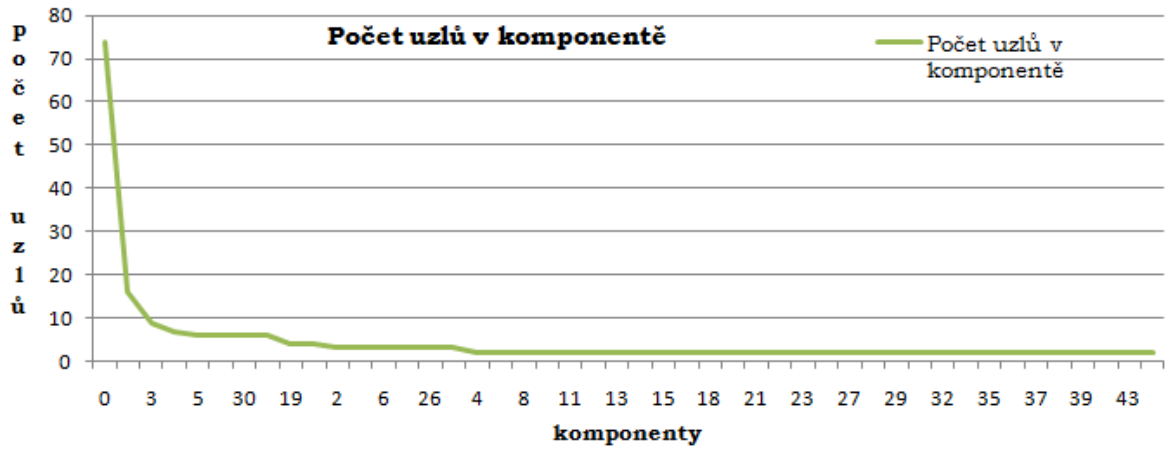
Výsledný graf vznikl stejnou cestou jako v experimentu popsaném v podkapitole 8.1 a to cestou transformace bipartitního grafu do unipartitního grafu. Také byl vytvořen příslušný objekt *Graph* s vrcholy, hranami a jejich vlastnostmi.

Jako výsledek vznikl graf o 216 vrcholech. Celkový počet hran je 762. Tento graf byl uložen ve všech formátech (graphml, csv pro gephi, jednoduchá reprezentace grafu v csv a xml).

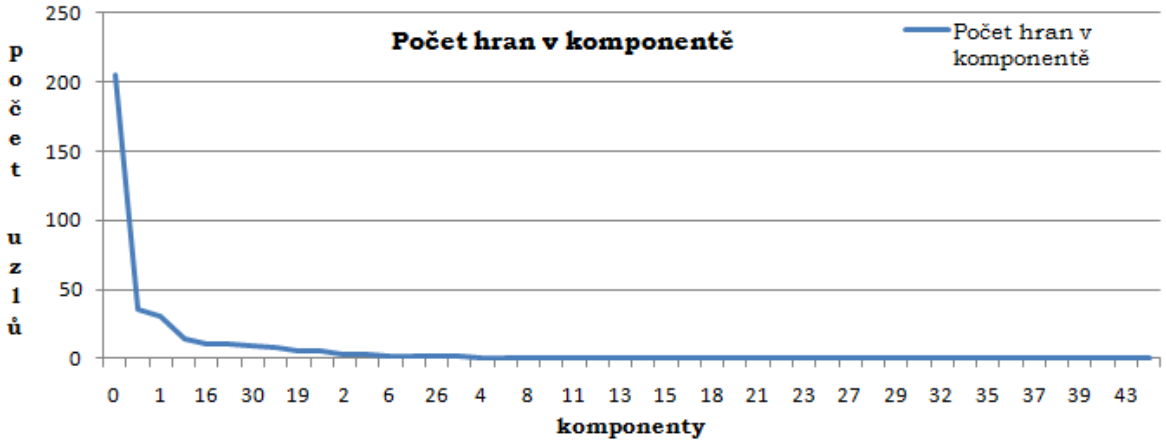
Následně byl graf zobrazen v nástroji Gephi. Hrany a vrcholy byly naimportovány do tohoto nástroje zvlášť pomocí csv souborů vytvořených před tím. Pomocí některých vnitřních metod Gephi bylo vylepšeno zobrazení grafu sítě. Obrázek se zobrazením je v příloze B.

Krok 3

V grafu, který vznikl v předchozím kroku, byly nalezené všechny souvislé komponenty, které byly následně uložené jako samostatné grafy do objektů typu *Graph*. V tomto grafu bylo nalezeno celkem 46 souvislých komponent různé velikosti. Výsledky uvádíme na obrázcích 8.2.1, 8.2.2 a v tabulce 8.2.1.



Obrázek 8.2.1: Diagram počtu uzlů v komponentách



Obrázek 8.2.1: Diagram počtu hran v komponentách

počet uzlů	počet komponent
2	30
3	6
4	2
6	4
7	1
9	1
16	1
74	1

počet hran	počet komponent
1	30
2	4
3	2
6	2
8	1
9	1
11	2
14	1
31	1
36	1
205	1

Tabulka 8.2.1: Tabulka počtů komponent dle počtů uzlu v nich

Největší z nalezených komponent je komponenta o 74 vrcholech, propojených mezi sebou 205 hranami. Další komponenty jsou mnohem menší, dokonce většina z nich jsou komponenty o dvou vrcholech. Průměrný počet uzlů v komponentech je 4,695652 a průměrný počet hran je 8,2826087.

Komponenty lze rozdělit dle struktury, za použití navrhnutých typů rozdělení v minulem experimentu, s výjimkou však toho, že jako minimální váha se nebude brát 1 ale 2.

Typ struktury	**Počet komponent**
Dvojice slabá	23
Dvojice silná	2
Trojice – přímka	4
Trojice – hvězda	2
Hvězda klasická	2
Hvězda s podskupinou	1
Hvězda plus jedinec.	1
Dvě hvězdy	-
Jiný	6

Tabulka 8.2.2: Rozdělení komponent dle struktury

9. Závěr

V této práci byl proveden rozbor pojmu sociální síť a průzkum existujících metod vytvoření a analýzy sociálních sítí. Byly provedeny experimenty na datech stažených ze systému pro sdílení zdrojových kódu www.codeplex.com. Nad těmito daty byl vytvořen bipartitní graf a následovně provedena jeho transformace do unipartitního grafu uživatelů. Ve výsledném grafu byly nalezeny souvislé komponenty, na kterých se následně ukázala analýza vztahů a rolí s pomocí výpočtů a vyhodnocení významu grafových centralit Degree, Closeness, Betweenness a Eigenvector. Dále byla naimplementována a otestovaná metoda filtrování sítě, cestou eliminace nevýznamných vazeb mezi uživateli. Nad výslednými grafy a jejich komponentami se provedla řada experimentů stejně jako nad celkovým grafem před eliminací.

Po provedení experimentů a vyhodnocení jejich výsledků jsme schopni tvrdit, že námi zvolené metody vytvoření a analýzy grafu sociální sítě jsou přínosem hlavně pro analýzu komunit a týmů, které nemusíme ve zdrojových datech přímo odhalit. Takovéto komunity můžeme nalézt nejen v projektu CodePlex, ale i v řadě dalších systémů určených nejen pro komunikaci a spolupráci lidí v internetové síti. V odlišnosti od jiných metod analýzy, o kterých bylo zmíněno v kapitole 5.2, je ve schopnosti odhalit vlastnosti komunit a jich účastníků na základě vytvořené sociální sítě.

9.1 Další možnosti rozšíření práce

V průběhu studia problematiky a provádění experimentů byly nalezeny i další metody, které by mohly doplnit algoritmy, pomocí nichž byly v této práci provedeny experimenty. Do stávajícího algoritmu je možné doplnit logiku vyhodnocení počtů nebo typu aktivit účastníků na projektech a použít tyto informace třeba k vytvoření grafu sítě jiného typu. Některá rozšíření již byly naimplementované a popsané v kapitole 7 (fáze 1a a 1b). Bylo to rozšíření vytvoření grafu dle časové složky a dle typu aktivit uživatelů. Obě rozšíření je možno kombinovat dohromady a tím vytvářet nové grafy s jinými vlastnostmi a dalším přínosem pro analýzu sítě.

V této práci jsme se zabývali analýzou neorientovaného grafu, což ale není podmínkou pro sociální síť. Pomocí analýzy nejen společných projektů uživatelů sítě, ale i analýzou intenzity a druhu aktivit každého z účastníků je možné vytvořit orientovaný graf uživatelů (motivováno práci Van Der Aalst [22] . Analýza orientovaného grafu by mohla přinést nové a zajímavé výsledky z pohledu celé sítě a také o každém účastníkovi zvlášť.

Pro potřebu vzniku menšího a přehlednějšího grafu sítě, byla v druhém experimentu (viz. Kapitola 8.2) znázorněná metoda redukce grafu, cestou eliminace nevýznamných vazeb mezi uživateli. Jiná možnost rozšíření této metody, může být i metoda eliminace aktivit účastníků, a to nejen dle počtu ale i dle typu aktivity. Cílem by bylo vytvoření grafu nesoucího další vlastnosti a jiné zajímavé podklady pro analýzu komunit a jejich částí.

Pro analýzu komponent výsledného grafu byly zvolené základní grafové centrality. Tuto analýzu je možno rozšířit o rozbor dalších grafových indexů, a to například cluster koeficient pro nalezení komunit a provedení experimentů pro nalezení shluků v celém grafu, například pomocí techniky Fuzzy K-Means [3] .

Dalším přínosem je studium fenoménu Power Law, který se poměrně často vyskytuje v sociálních sítích a jeho vazby na námi generovanou síť z projektu CodePlex [23] .

Literatura

[1] Everett Stiles and Xiaohui Cui: Workings of Collective Intelligence within Open Source Communities, Advances in Social Computing, Lecture Notes in Computer Science, 2010

[2] Fiedler: Speciální matice a jejich užití, SNTL, Praha, l980

[3] Heiko Timm, Christian Borgelt, Rudolf Kruse: Fuzzy Cluster Analysis with Cluster Repulsion, 2002

[4] Heng-Li Yang a Jih-Hsin Tang: Team structure and team performance in IS development: a social network perspective, Information & Management, Volume 41, Issue 3, January 2004, Pages 335-34

[5] Hliněný, Petr: Základy teorie grafů, 2010

[6] Iacopo Carreras, Daniele Miorandi: Eigenvector Centrality in Highly Partitioned Mobile Networks: Principles and Applications, 2007

[7] Jin Xu, Yongqin Gao, Scott Christley, Gregory Madey: A Topological Analysis Of The Open Source Software Development Community, 2005

[8] Jin Xu, Scott Christley, and Gregory Madey: Application of Social Network Analysis to the Study of Open Source Software, 2006

[9] Kateřina Sabová, Sociální skupiny, 2003

[10] Kazuya Okamoto, Wei Chen: Ranking of Closeness Centrality for Large-Scale Social Networks, 2008

[11] Lenka Buštíková: Analýza sociálních sítí, Sociologický časopis, 1999, Vol. 35 (No. 2: 193-206)

[12] M. E. J. Newman: Networks: An Introduction, 2010

[13] Národní ústav odborného vzdělávání a občanské sdružení Projekt Odyssea: Týmová práce, říjen 2006

[14] Peyman Nasirifard, Vassilios Peristeras: Extracting and Utilizing Social Networks from Log Files of Shared Workspaces, 2009

[15] Scott Christley, Greg Madey: Collection of Activity Data for SourceForge Projects, 2005

[16] Subhajit Datta, Vikrant Kaulgud, Vibhu Saujanya Sharma, Nishant Kumar: A social network based study of software team dynamics, India Software Engineering Conference, 2010

[17] Tim O'Reilly: What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Softwar, 2005

[18] Tomáš Marek: Sociální sítě, aktéři a instituce v regionálním rozvoji, 2003

[19] URL: http://www.community.cz/

[20] URL: http://socialnisite.cz/

[21] Ulrik Brandes: A Faster Algorithm for Betweenness Centrality, 2001

[22] Van Der Aalst, Wil M. P. and Reijers, Hajo A. and Song, Minseok: Discovering Social Networks from Event Logs, 2005

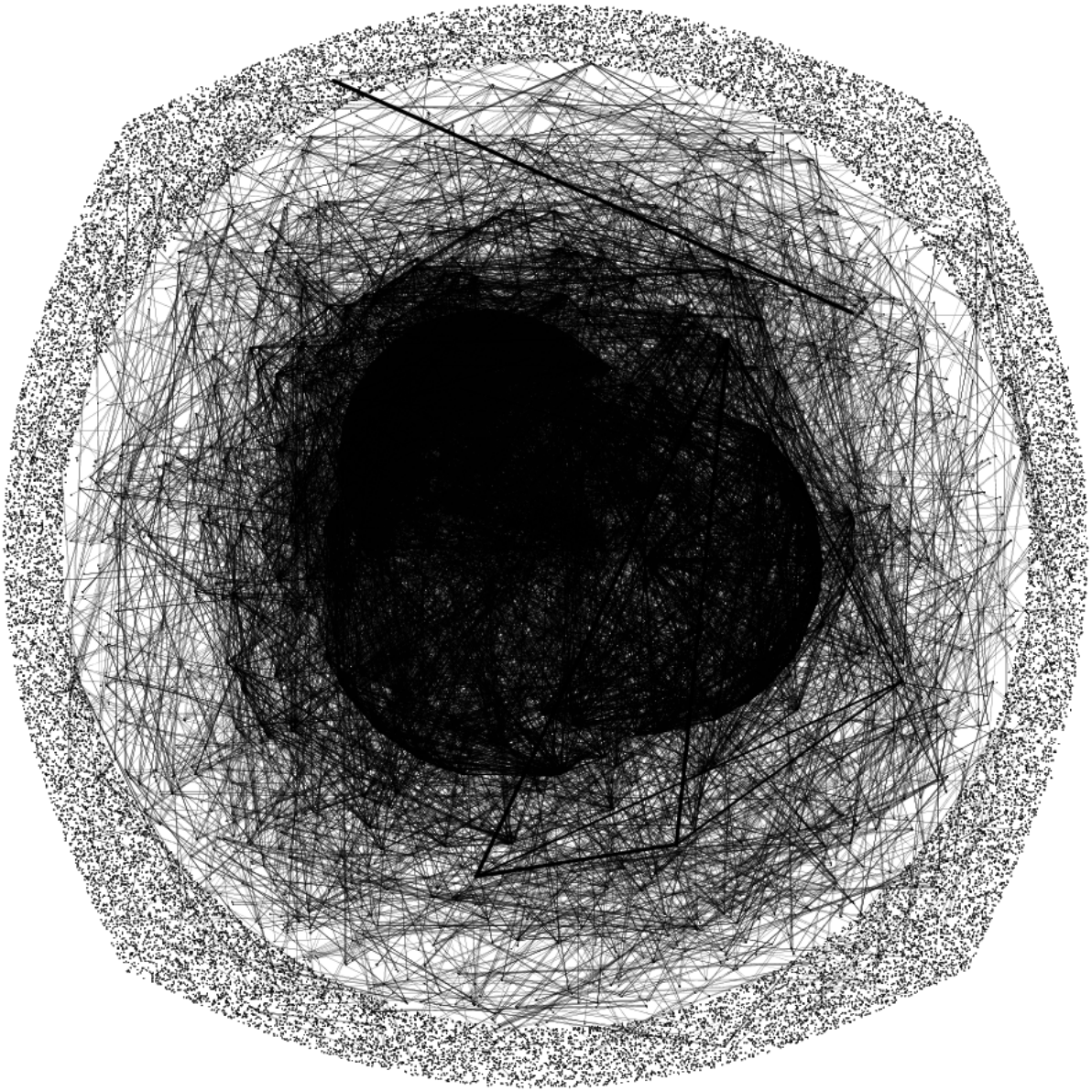
[23] William Aiello, Fan Chung, Linyuan Lu**:** A Random Graph Model for Power Law Graphs, 2001

[24] Zdeněk Molnár: Možnosti využití sociálních sítí v Competitive Intelligence, 2010

[25] Yongqin Gao, Greg Madey: Network Analysis of the SourceForge.net Community, 2007

Přílohy

A. Graf síti uživatelů CodePlexu (váha od 1).



B. Graf síti uživatelů CodePlexu (váha od 2).